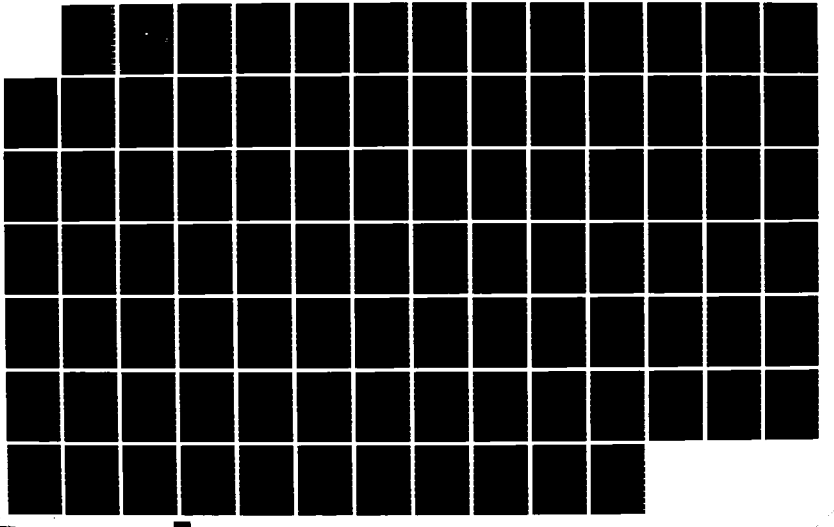
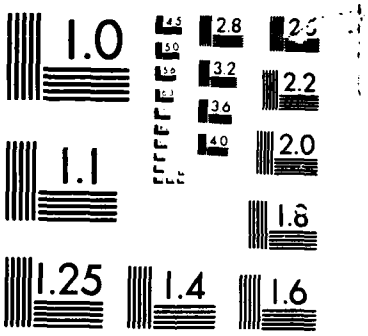
AD-A168 909 A SOLID STATE DATA RECORDER FOR SPACE-BASED APPLICATIONS USING MAGNETIC BUBBLE MEMORY(U) NAVAL POSTGRADUATE SCHOOL MONTEREY CA T D'ERCOLE MAR 86 1/1

UNCLASSIFIED F/G 14/3 NL




1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.8
3.2
3.6
4.0

AD-A168 909

NAVAL POSTGRADUATE SCHOOL
Monterey, California

DTIC
ELECTE
JUN 2 3 1986
S D
E

THESIS

A SOLID STATE DATA RECORDER
FOR SPACE-BASED APPLICATIONS
USING MAGNETIC BUBBLE MEMORY

by

Tina-Marie D'Ercole

March 1986

Thesis Advisor: R. Panholzer

Approved for public release; distribution is unlimited.

DTIC FILE COPY

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S)	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION Naval Postgraduate School 6b. OFFICE SYMBOL (If applicable) 62	7a. NAME OF MONITORING ORGANIZATION Naval Postgraduate School
6c. ADDRESS (City, State, and ZIP Code) Monterey, California 93943-5100	7b. ADDRESS (City, State, and ZIP Code) Monterey, California 93943-5100
8a. NAME OF FUNDING/SPONSORING ORGANIZATION 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (City, State, and ZIP Code)	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. / PROJECT NO / TASK NO / WORK UNIT ACCESSION NO.

11. TITLE (Include Security Classification)
A SOLID STATE DATA RECORDER FOR SPACE-BASED APPLICATIONS USING MAGNETIC BUBBLE MEMORY

12. PERSONAL AUTHOR(S)
Tina-Marie D'Ercole

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Master's Thesis	FROM _____ TO _____	1986 March	91

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Bubble Memory; Magnetic Bubble Memory; Digital Data Recorder; Space Data Recorder; NASA Get-Away Special Program; Solid State Data Recorder

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

Earthbound recording systems come in many sizes and use various mediums on which to record the data. The harsh environment of space, however, introduces some unique problems. This limits the number of choices not only for the type of system but also for the optimum recording medium. How changes in temperature, radiation, lack of air, etc., affect the performance of the device must all be considered.

Magnetic bubble memory technology implemented in a solid state recorder is a possible solution. Included in this thesis is a description of the development and history of the magnetic bubble memory, along with a comparison to other technologies. The design and implementation of a digital data recorder using off-the-shelf four-megabit devices is presented. A schematic of the data recorder and software used is included in the appendices.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT	21. ABSTRACT SECURITY CLASSIFICATION
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS	UNCLASSIFIED
22a. NAME OF RESPONSIBLE INDIVIDUAL Rudolf Panholzer	22b. TELEPHONE (Include Area Code) 408-646-2154 22c. OFFICE SYMBOL 62Pz

DD FORM 1473, 84 MAR 83 APR edition may be used until exhausted
All other editions are obsolete

SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

Approved for public release; distribution is unlimited.

A Solid State Data Recorder for Space-based Applications using Magnetic Bubble Memory

by

Tina-Marie D'Ercole
Lieutenant, United States Navy
B.S., United States Naval Academy, 1980

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
March 1986

Author: Tina-Marie D'Ercole
Tina-Marie D'Ercole

Approved by: Rudolf Panholzer
Rudolf Panholzer, Thesis Advisor

Mitchell L. Cotton
Mitchell L. Cotton, Second Reader

Harriett Rigas
Harriett Rigas, Chairman, Department of Electrical and Computer Engineering

John Dyer
John Dyer,
Dean of Science and Engineering

ABSTRACT

Earthbound recording systems come in many sizes and use various mediums on which to record the data. The harsh environment of space, however, introduces some unique problems. This limits the number of choices not only for the type of system but also for the optimum recording medium. How changes in temperature, radiation, lack of air, etc., affect the performance of the device as a whole must all be considered.

Magnetic bubble memory technology implemented in a solid state recorder is a possible solution. Included in this thesis is a description of the development and history of the magnetic bubble memory, along with a comparison to other technologies. The design and implementation of a digital data recorder using off-the-shelf four-megabit devices is presented. A schematic of the data recorder and software used is included in the appendices.

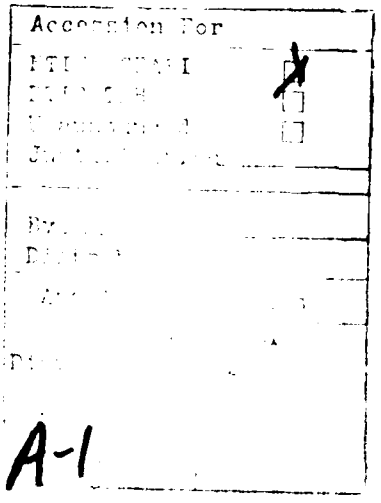


TABLE OF CONTENTS

I. INTRODUCTION 8

II. BUBBLE MEMORY CHARACTERISTICS 10

A. MAGNETIC BUBBLE MEMORY PERFORMANCE 10

B. COMPARISON OF MBM TO OTHER MEMORY DEVICES . . 12

III. MAGNETIC BUBBLE DEVELOPMENT 16

IV. FUNCTION OF SUPPORT CIRCUITRY 24

V. DESIGN OF THE SSDR 33

A. PROTOTYPE 33

B. SOLID STATE DATA RECORDER (SSDR) 36

1. General Purpose SSDR 38

2. Adaptation of the SSDR to the NPS GAS Experiment 39

3. Limitations and Alternative 41

APPENDIX A: SCHEMATIC OF THE SOLID STATE RECORDER . . . 43

A. POWER SWITCHING CARD (ECARD) 43

B. CONTROLLER CARD (MICROPROCESSOR/DMA) 47

APPENDIX B: SOFTWARE CODE 53

LIST OF REFERENCES 88

BIBLIOGRAPHY 89

INITIAL DISTRIBUTION LIST 90

LIST OF TABLES

I DESIGN PARAMETERS 9

LIST OF FIGURES

3.1 Magnetic Domains Under Magnetic Bias Field 17
3.2 Magnetic Bubble Unit Assembly 18
3.3 Movement of Bubbles Under Chevrons 20
3.4 Organization of Bubble Memory (One-Half Chip) . . . 21
3.5 Seed Bubble and Bubble Generation 22
3.6 Swapping and Replication Configuration 23
4.1 System Block Diagram 25
4.2 CPD Logic Diagram 26
4.3 Ten Functional Blocks of the Controller 27
4.4 Six Parametric Registers 29
4.5 Parametric Register Organization 30
4.6 Commands Used by Bubble Memory Controller 31
5.1 Generic Data Recorder 37
5.2 Implementation of Hardware Comparator 38
5.3 Basic Channel Configuration 40

ACKNOWLEDGEMENTS

Since every person you come into contact with contributes something to your life, no matter how small, I would like to take this opportunity to give my thanks to the following individuals. I believe no one can accomplish anything without the help and support of friends.

1. Professor Panholzer for his direction and understanding. He exemplifies what the word teacher really means.
2. Roger Jaskot, Taylor Tripp, George Johnston (and Maureen), Debbie Anderson, Beth Patridge, Don Gold (and Michele), Elizabeth Sternaman, and Ann Stencil for their patience, support and friendship.
3. Tom and Holly Frey for their help and support. Tom is one excellant briefing partner! Holly is one special friend.
4. Elaine Christian, the heart of the "EE" curricular office and a very special person.
5. The "Bust Your Budget A Month Club" for adding to the fun of the non-academic hours!
6. Bill Cavitt, Nancy and Nels (DH) Petersen, Cathi and Wayne Borchers for making the last quarter a memerable one and helping to celebrate the completion of this thesis!
7. David Rigmaiden for his technical expertise and help with the Shuttle Project as a whole.
8. My oldest and dearest friend, Rick Cottrell, for his friendship and confidence in me since we were children. His sensitivity and insight is priceless.
9. Finally, John and Emily D'Ercole, my three sisters, Marianne, Elizabeth, and Angela, and their families for their belief in me as a person and support ever since I can remember.

I would also like to take this opportunity to remember the seven astronauts who lost their lives 28 Jan 1986, when the Space Shuttle Challenger exploded. We have come a long way in exploring this frontier we call "space." There is such a tremendous risk involved and we, as individuals, owe a great deal to those who are willing to take that risk.

I. INTRODUCTION

"Space, the final frontier . . . ."[1] For centuries space has been an unknown frontier to man, a frontier that has been, and is still, studied in great detail. Man, in his quest for knowledge, has progressed from performing experiments in space with unmanned vehicles to manned reusable platforms that can remain in space for days. With the birth of the Space Transportation System (STS or Shuttle), an opportunity has been provided for persons outside the National Aeronautics and Space Administration (NASA) and military organizations to send their own experiments into space. This opportunity is in the form of a program that is known as the Get Away Special Program or GAS.

The GAS program provides guidelines for the designers of an experiment. These guidelines include such tips as the recommended use of the standard GAS container provided by NASA, electronic interfaces to the Shuttle cargo bay, safety tips, and environmental considerations. Table I provides the conditions under which the payload must perform [Ref. 1: p. 2].

An experiment that will be flown on the Shuttle by students from the Naval Postgraduate School (NPS) is designed to measure the vibro-acoustic power levels that occur in the forward one-third of the STS cargo bay during launch. Acoustic vibrations may cause widespread damage to the electronic equipment. Three microphones will be used to detect the acoustic noise levels. The data obtained will be stored in a recorder. Since the required data will be produced during the first two to three minutes of launch, (the recorder will be idle for the remainder of the flight),

---

[1]This quotation was taken from the television series "Star Trek."

TABLE I

DESIGN PARAMETERS

	MAXIMUM	MINIMUM
Altitude (NM)	400	120
Percent of time in earth's shadow	40	32
Temperature (^{0}C)		
Prelaunch	5	50
Launch	5	65
On-orbit	-160	100
Entry/Postlanding	-45	105
Typical overall temp		+65 to -50
Pressure (atm)	10^{9}	10^{14}
Steady-state acceleration (g)	10^{2}	10^{7}
Acceleration in maneuvers (g)	0.1	0.001
Random Vibrations	.125 g^2/Hz between 80-1000 Hz	

the memory needed to record the data must be rugged, reliable, and non-volatile. It must be able to withstand the impact of lift-off and the hazards of a harsh environment such as space.

A solid state magnetic bubble memory data recorder (SSDR) has been chosen to perform the task of data storage. This and all other components of the whole experiment have been designed with the previous parameters as the specifications. This thesis presents the development of the magnetic bubble memory and its implementation as the recording medium in the solid state data recorder.

II. BUBBLE MEMORY CHARACTERISTICS

Equipment used in experiments performed in space must be inherently rugged in order to withstand the severe environmental surroundings. What may be a successfully operating piece of equipment on earth may have to be upgraded in certain areas in order to function reliably in space [Ref. 2: p. 1]. The first section of this chapter is devoted to a discussion of the performance of the Magnetic Bubble Memory (MBM) in such a hazardous environment. The remaining section compares it with other memory devices when required to function as a data recorder.

A. MAGNETIC BUBBLE MEMORY PERFORMANCE

In space, exposure to intense radiation and temperatures that far exceed those experienced on earth are encountered. Studies have been conducted to determine just how severe these effects may be on the MBM. These studies have shown that it is virtually unharmed by exposure to neutrons, protons and gamma rays. Experiments have demonstrated that levels above 10^{15} neutrons/cm^2 do not cause permanent measurable changes in the properties of the materials used [Ref. 3: p. 2]. The MBM has also been exposed to transient radiation, such as X-rays. Even in the absence of shielding, tests have shown that failure occurred at a value of 5×10^6 rads/sec [Ref. 4: p. 4847]. This information can be put into perspective when one considers the radiation received by an already "space-born" system such as the Global Positioning System(GPS). It is in an orbit in which the satellites experience the greatest radiation dose to date by long-lived, earth-orbiting systems. Tests have shown that over a 448-day period, the average dose of radiation was 404 rads/day [Ref. 5: p. 477]. Radiation, therefore, will have very little effect on the MBM module.

Due to the lack of atmosphere in space the temperature range can be extensive. There are two major limitations on the operating temperature range of the MBM. The first limitation is in connection with the bias field. The second limitation is in connection with the rotating magnetic field. (Implementation of the two fields is discussed in Chapter 2.)

The permeability of both the bias field and the thin film of magnetic material on which the bubbles are formed is affected by varying temperatures. The composition of each part changes in accordance with changes in temperature, and thereby limits the operating temperature of the MBM. The effect on the rotating magnetic field is such that, as temperature decreases, the rotating field must increase in order to drive the bubbles through the medium.

Two other effects exist that may limit the operating temperature range of the MBM. First, to generate new bubbles--i.e., a write operation--a current pulse, which will be explained in the following chapter, is required. This pulse varies with changes in temperature. To ensure proper generation of each bubble, the pulse must be adjusted or else improper data may be recorded. The second effect occurs in a read operation. During the read cycle, the data is sent under a type of detector that senses the presence or absence of a bubble. This detector output changes with temperature. Since the bubbles' magnetic permeability and the detector's sensing ability vary with temperature, the operational temperature of the MBM is limited to a fixed range of values. Current MBM modules operate over a temperature range of -20^0 to $+85^0$ [Ref. 6: pp. 83-88].

Data already recorded in the MBM can be stored reliably over a full range of temperature from -40^0 to $+90^0$ [Ref. 7: p. 3]. The largest temperature restrictions imposed on equipment are those encountered in the military. Many of

the military applications require successful operation at temperatures beginning as low as -55° C. Research is in progress to improve the operating temperature capability of the MBM [Ref. 6]. For the GAS experiment, however, the advertised operating temperature range of the MBM is sufficient.

The MBM has no mechanical moving parts (see Chapter 2). With cassette recorders that require mechanisms to turn the recording medium, mechanical breakdown, corrosion, and slipping due to vibration may occur. None of these situations which may prove fatal to the mission, i.e. loss of data, will occur with the MBM.

B. COMPARISON OF MBM TO OTHER MEMORY DEVICES

A comparison of the MBM with other memories demonstrates further why using it as a data recorder in the space environment constitutes a practical choice. Memories can be separated into two basic categories: semiconductor and magnetic. Examples of the former include ROM, PROM, EPROM, E^2 ROM, RAM, and CCD. Examples of the latter include tape, disk, core and MBM. The semiconductor group uses voltage, charge or current levels to represent data. The magnetic group uses variation in magnetic flux [Ref. 8: p. 8].

The individual features of each memory should be considered in choosing an appropriate device; using one in space adds even more constraints to the selection process. Some required features, besides radiation hardness and temperaure range, are versatility, non-volatility, reliability, small size, low power consumption and, finally, cost.

When versatility is considered, some of the aforementioned memory devices can be eliminated. The PROM (programmable ROM) and EPROM (erasable programmable ROM) must be programmed by the user outside the circuit. While the PROM can be programmed only once, the EPROM has the capability of being programmed numerous times. In order to accomplish

this, however, it must be removed from the circuit and exposed to ultraviolet light. As a result, using either as a recording medium would be impractical.

The electronically erasable ROM, E^2 ROM, can be changed right in the circuit. Any byte can be erased in approximately 9ms (Intel E^2 ROM 2816A) without affecting the data in any other location. The entire memory can be erased in 9ms also [Ref. 9: p. 5-83]. The available chip sizes, however, do not compare to the amount that can be stored in one 1 Mbit MBM module or the 4 Mbit device. The E^2 ROM functions best as a read only memory suited for storing fixed programs, logic functions or code converters with the added capability of remote firmware update of program code and dynamic parameter storage [Ref. 9: p. 5-10].

Other memories can be eliminated when non-volatility is considered. If a memory is non-volatile, it has the ability to retain already stored data should a power failure occur. The semiconductor RAM (random access memory) is a volatile memory. Data recorded prior to a power failure would be lost.

The CCD, or charge-coupled device, was developed in an effort to find a semiconductor equivalent to the MBM. It is dynamic, and data must be internally shifted at a minimum rate or else electrons can be thermally generated, thereby modifying existing stored data [Ref. 8: p. 313]. Unless batteries are used to keep the RAM and CCD memories powered up in a stand-by mode, the data would be lost. The length of time the stand-by power is required rests on the mission of the Shuttle and the time it would take to return the experiment to the NPS. Adding stand-by power adds batteries and, consequently, adds to the weight and space situation inside the GAS cannister.

Tape recorders are frequently used because of their small size, comparatively low cost, and large data storage

capability that permits parallel storage of data on several channels. The tape, however, can be damaged should it be stretched or knocked off its tracks during launch. Tape also requires a very thin layer of air between itself and the recording heads. Finally, the moving parts aid in decreasing the overall reliability.

Magnetic disks and drums pose problems that are similar to the tape recorder's. Once again moving parts decrease the reliability. Another limiting factor is the amount of space available inside the GAS canister. Disks and drums require a great deal of space which leaves less room for other essential equipment.

Core memory has been around for many years. It is non-volatile and reliable, but size and power constraints limit its use as a piece of space equipment. In a self-contained experiment such as this one, power is critical. It is desirable to have nominal power consumption. As the amount of data to be stored increases, so does the power consumption, size, and cost of the core memory--to a much larger degree than with other memories. Should the design of this recorder be such that all of the MBM cards be turned on at once, it too would require a great deal of power. However power switching becomes a viable option when implementing the recorder with the MBM. Thus, the only active MBM card needs to be the one currently in use, thereby conserving a great deal of power.

The preceding discussion of the MBM's ability to withstand conditions in space, and the comparison with other memory devices, helps to show why--in addition to its being versatile and non-volatile--the MBM is the best device for the GAS experiment. Even though there is shielding around the device to protect against radiation, choosing a device that is inherently unaffected by radiation provides an added element of reliability. Although the operating temperature

range of the MBM is not as extensive as would be required in several military applications, it is adequate for the purposes of the experiment. The MBM is reliable because it has no mechanical parts that move. The specific device chosen for the GAS experiment is the Intel Corporation Magnetic Bubble Memory 7114.

III. MAGNETIC BUBBLE DEVELOPMENT

The MBM has been in existence for approximately twenty years. After its introduction by Andrew Bobeck--an employee of Bell Labs--many companies became involved in the study and manufacture of bubble memory devices. They soon discovered, however, that the MBM required complicated control circuitry. It was also difficult to interface with existing hardware. As a result, the decision to discontinue production of the MBM was not long in coming [Ref. 10: pp. 30-32]. The two major manufacturers remaining are Intel Corporation and Hitachi.

A study of the technology used in producing and manipulating "bubbles" is important for a better understanding of the MBM's overall operation and its use in the SSDR. This chapter is devoted to an explanation of this technology, while Chapter 3 explains how the support chips of the MBM function. Before beginning this chapter, however, it is important to point out that all figures in this chapter and the following chapter represent the 1 Mbit bubble device. The GAS experiment is using the 4 Mbit device. However, the basic operation is the same. Intel's 4 Mbit device "uses the same architecture as the 1 Mbit device. It has 8 identical sections instead of 4 and each section is enlarged to store double the number of bubbles. The result is a fourfold increase in capacity" [Ref. 9: p. 6-227].

The bubble chip of Intel Corporation's 7114 MBM begins with the formation of a nonmagnetic garnet wafer on top of which a thin ferromagnetic film is deposited. Ferromagnetism causes the atoms of a material to align with parallel magnetic orientations. If the material is thick enough, the orientation of the groups of atoms, or domains as they are called, occurs in three dimensions. In the case

of the MBM, however, the film is very thin--in practice less than 1/1000 inch thick. This thinness restricts the domains to two directions, one perpendicular to the substrate, the other parallel. Unless a magnetic field is present, these domains have a snakelike shape. As a magnetic field perpendicular to the film is applied, the snakelike structures begin to shrink. The field is then increased to a point where the domains have a cylindrical appearance with a three- micrometer diameter. These domains, when observed through a microscope, look like a circle or "bubble"--from which the name Magnetic Bubble Memory is derived (see Figure 3.1). The presence of a bubble is a binary 1; the absence, a 0 [Ref. 9: p. 6-3].

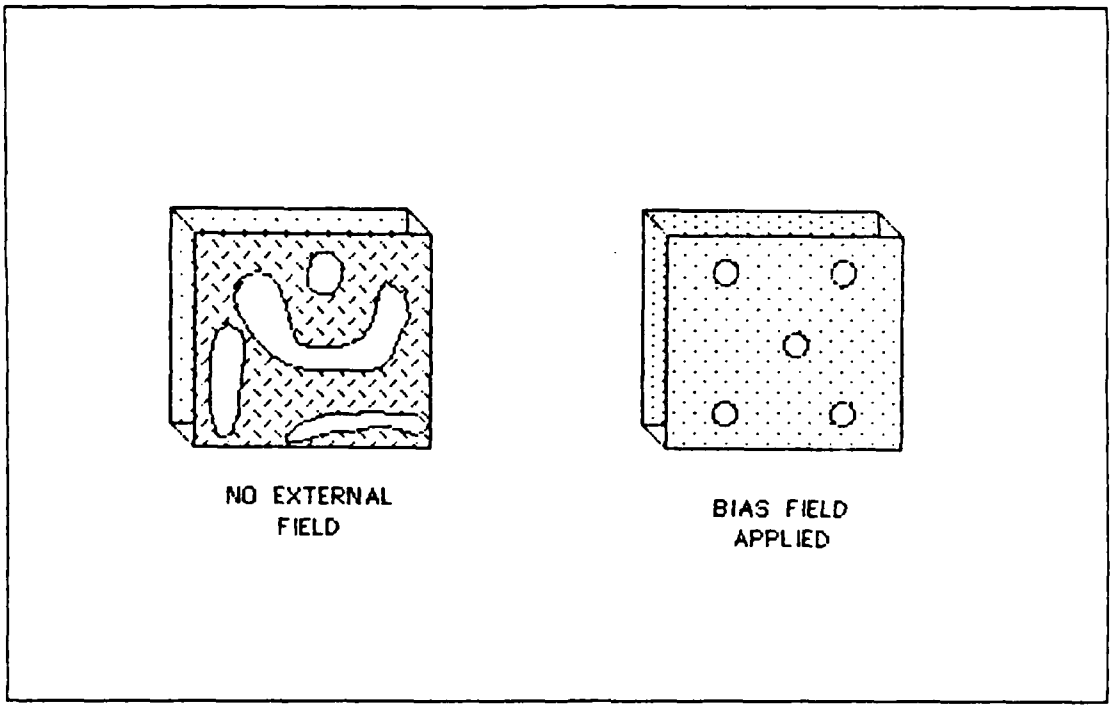


Figure 3.1 Magnetic Domains Under Magnetic Bias Field.

The fixed field that maintains the shape and stability of the bubbles is known as the bias field. This bias field is produced by two permanent magnets positioned on each side

of the MBM substrate. They remain unaffected by any type of power fluctuation. As a result, the integrity of the data that the bubbles represent is maintained, making the MBM a non-volatile memory device [Ref. 9: p. 6-4].

The MBM does not operate by moving the recording medium as is the case with the tape in a tape recorder or disks in a disk drive. In the MBM, the data is moved under the influence of a rotating magnetic field. The rotating field is generated by sending current through two coils which are wrapped perpendicular to one another around the substrate. Figure 3.2 shows how the different parts are assembled to produce the MBM module.

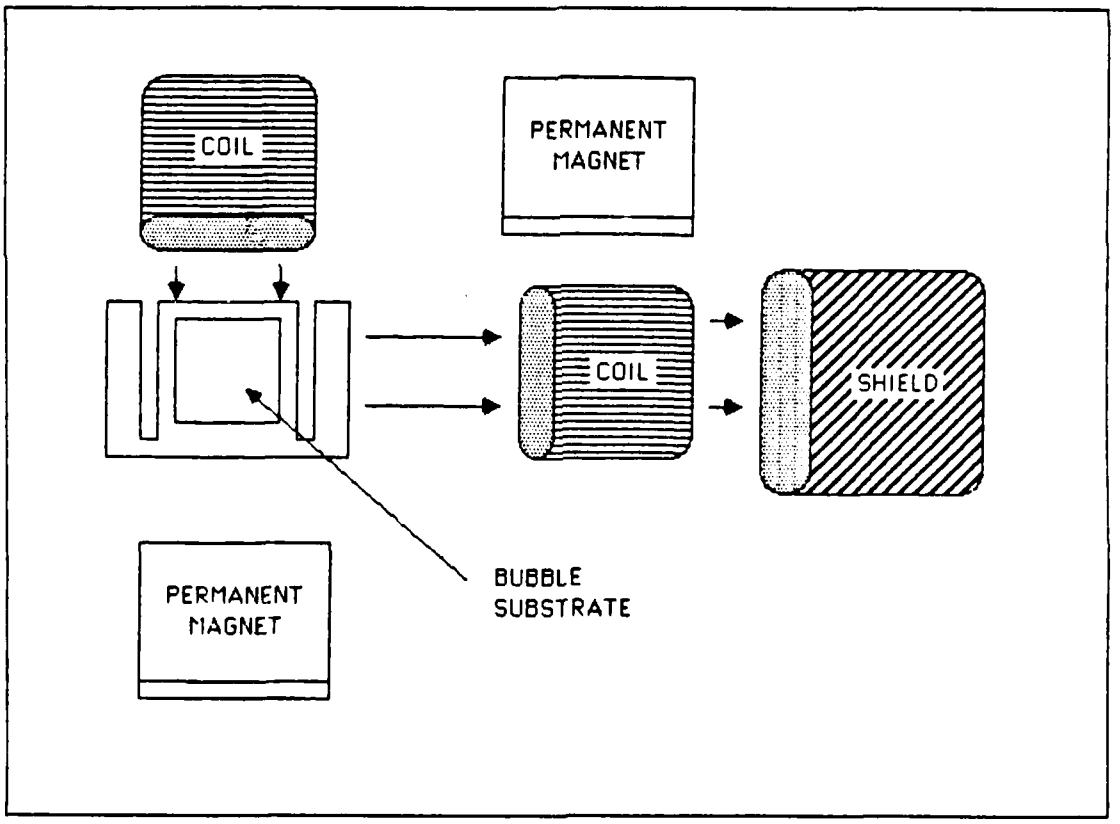


Figure 3.2 Magnetic Bubble Unit Assembly.

There is one more important feature that aids in the movement of the bubbles from one location to another. In order to control the direction of movement, a magnetic field gradient must be present. A magnetic film is deposited on top of the bubble substrate in the form of a number of paths. These paths have the shape of asymmetrical chevrons. Being asymmetrical, one side of the chevron has more energy than the other. The bubble will propagate toward the smaller side in the presence of the drive field (see Figure 3.3). The chevrons, in conjunction with the rotating magnetic field, provide the capability to move data from one location to another without moving the medium on which the data is stored. [Ref. 9: pp. 6-4 - 6-6].

The memory itself is designed in a block replicate architecture. The fundamental idea in this type of design is that the data is written in or read out in parallel from an input or output track, see Figure 3.4 (From the figure, note that there is a difference between the input and output tracks. The difference will be explained later in this chapter.)

The block-replicate architecture consists of a specified number of storage loops. An MBM has 320 loops, 48 of which are spares. An extra loop, called the bootloop, is used to keep track of the active loops and the spares. All 320 loops are divided into four groups or quads. This helps to shorten the read and write cycle times.

The quads are reduced even further to odd and even loops. The even loops store the even-numbered bits. Likewise, the odd loops store the odd-numbered bits. When a read operation is performed, the bits are interleaved back into the original order in which they were received and sent out serially on the output track. A write operation does not require this interleaving process, for it is a write operation that initially separates the bits to the odd and even loops.

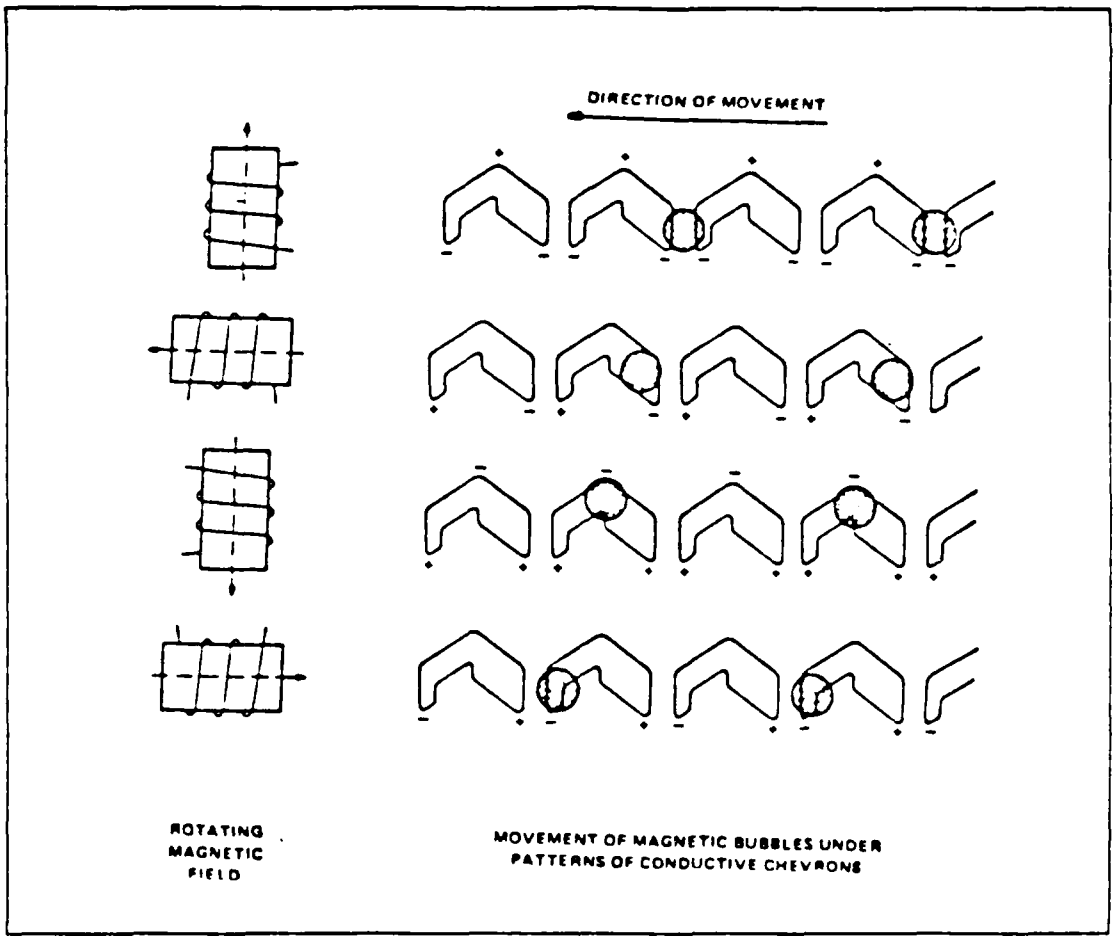


Figure 3.3 Movement of Bubbles Under Chevrons.

The input and output tracks are serial devices. A distinction exists between the two tracks because they perform completely different tasks. The input track performs a swapping function; the output track performs a replication function.

To better explain each process, bubble generation must first be understood. A seed bubble is always present at one end of the input track and is initially generated by an electric current pulse which splits a hairpin loop of conductive material. Because of the interaction between the drive field, bias field, and conductive material (permalloy

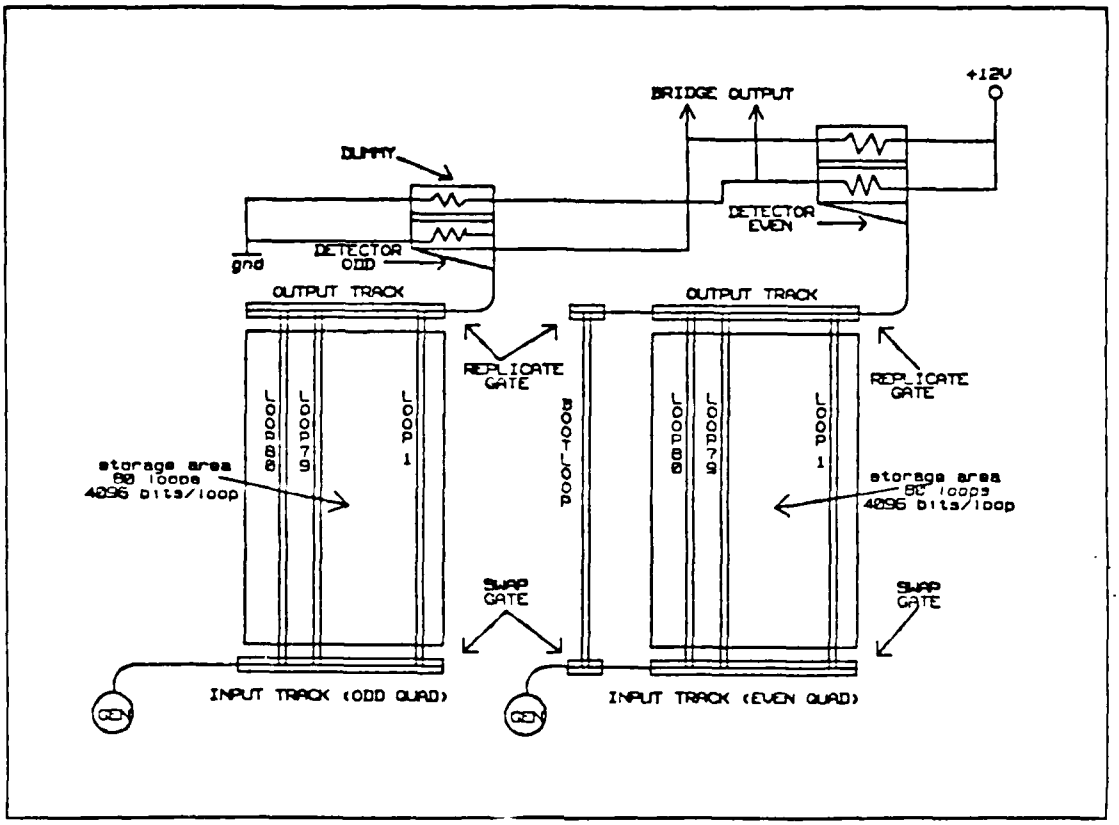


Figure 3.4 Organization of Bubble Memory (One-Half Chip).

patch, see Figure 3.5), this bubble maintains a kidneylike shape. Once created, it remains in existence for as long as the bias field does. When a binary 1 is to be generated, the seed bubble is split in two by the current pulse. One of the halves remains under the permalloy patch as the seed; the other is driven to the input track via the rotating field. To store a binary 0, the pulse is omitted.

As stated, the input track performs a swapping function. Once the bubble is generated and travels down the input track to the specific loop on which it is to be stored, another current pulse is generated. This pulse causes the new data to "swap" places with the old data on the storage loop, and the old data is destroyed.

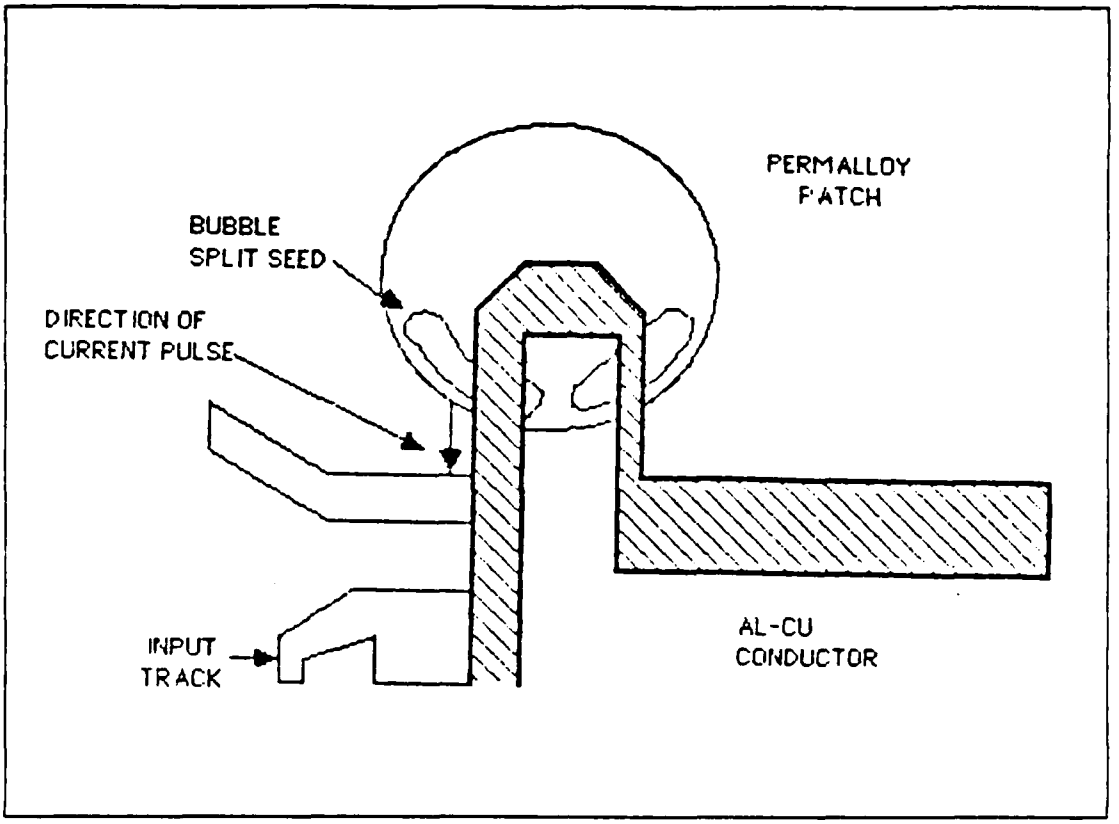


Figure 3.5 Seed Bubble and Bubble Generation.

The bubbles are replicated when a read operation is performed. Each bubble on a storage loop acts essentially as its own seed bubble. The bubble is transferred under a large element, where it is stretched out. A current pulse cuts the bubble in two, leaving one half to remain in memory and the other half to be read as output. These new bubbles travel down the output track serially through a bubble detector. Detection is accomplished by passing the bubble under a "bridge" of magnetoresistive material. When a bubble passes under the bridge, the resistance changes and slightly modulates the current through the bridge. This fluctuation is then translated to a one or a zero. After detection, the output bubbles are destroyed (see Figure 3.6) [Ref. 9: pp. 6-7 - 6-10].

As can be seen, an intricate design is used in the making of a magnetic memory module. Additionally, the MBM is never "spoken to" directly by the microprocessor. A group of support chips is used for communication between the two, and this is the topic of discussion in Chapter 3.

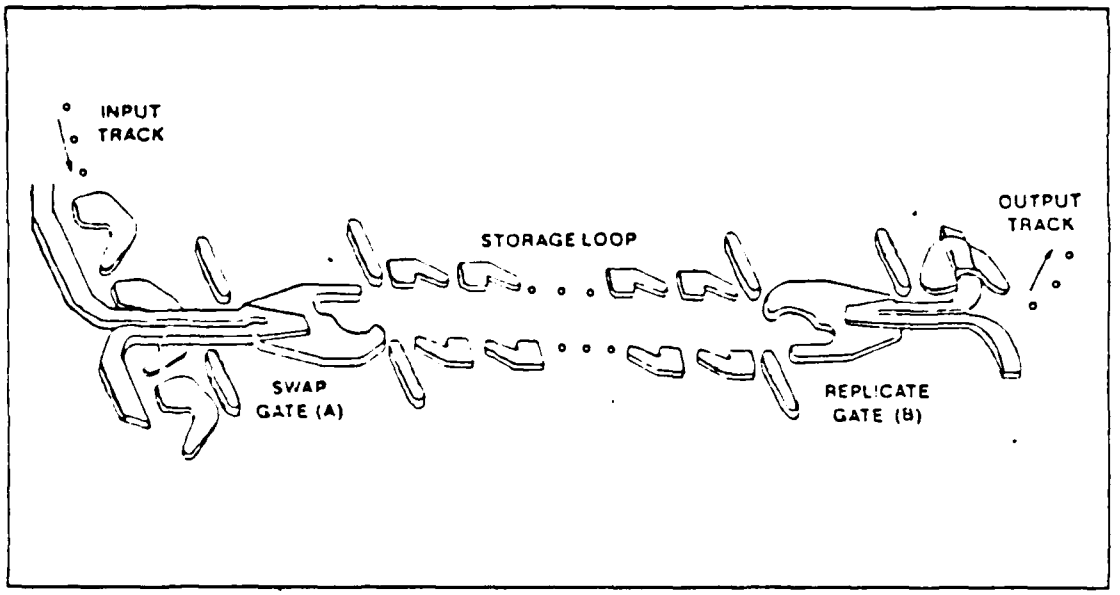


Figure 3.6 Swapping and Replication Configuration.

IV. FUNCTION OF SUPPORT CIRCUITRY

A difficult obstacle to overcome in the introduction of the MBM into the commercial market was interfacing it with existing hardware. Initially, the support chips required by the MBM were purchased separately. In the past few years, however, Intel has introduced a package consisting of the MBM and its requisite support chips. These chips are:

1. 7244 Formatter/Sense Amplifier (FSA),
2. 7234 Current Pulse Generator (CPG),
3. 7250 Coil Predriver (CPD),
4. 7224 MBM Controller (BMC).

The BMC, in conjunction with the FSA, CPG, and CPD, carry out all communication with the MBM. Figure 4.1 shows how the chips interface with one another.

The FSA is a dual formatter/sense amplifier that contains on-chip sense amplifiers, a full FIFO data block buffer, and error-detection and correction circuits. As explained in Chapter 2, the bubbles are sent under a magnetoresistive bridge during a read operation. If a bubble is detected, the resistance of the bridge changes. This is the signal that a bubble or a one is present. The sense amplifiers in the FSA perform a sample-and-hold function on this input signal thereby, producing a digital one or zero.

The FSA then formats the data in the following manner. As explained in Chapter 2, each MBM has an extra loop that is known as the bootloop. This extra loop contains the information pertaining to all the active and inactive loops in the bubble module. When the FSA receives an incoming data bit, it confirms that it is from an active loop within the MBM by referring to the bootloop register. If the bit is from an active loop, it is stored in the FIFO buffer. If it is from an inactive loop, it is ignored. The FIFO in the

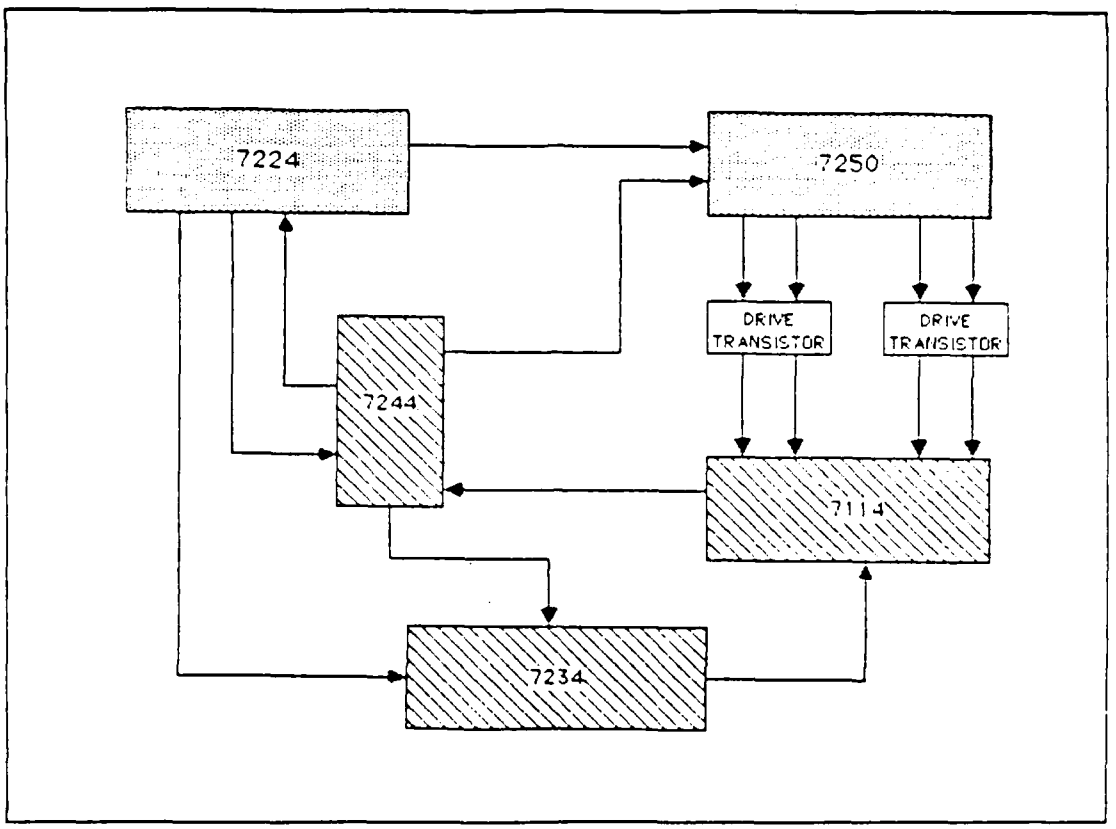


Figure 4.1 System Block Diagram.

FSA is a serial, first-in-first-out buffer that can hold 256 bits of data (272 without error correction). The data is then sent to the BMC. In the write operation, the FSA enables the current pulses of the CPG that cause the bubbles to be generated.

Various current pulses are used to generate a bubble as well as to replicate one. The CPG performs this function. The CPG also converts digital timing signals to analog current pulses suited to drive the MBM.

The CPD, along with the two drive transistors (see Figure 4.1), supply the drive currents for the rotating magnetic field. Four signals (positive and negative X and Y waveforms) are sent to the CPD from the MBM Controller. The appropriate durations and phases must be maintained in order

to control the rotating field that moves the bubbles (see Figure 4.2) [Ref. 9: pp. 6-11 - 6-12].

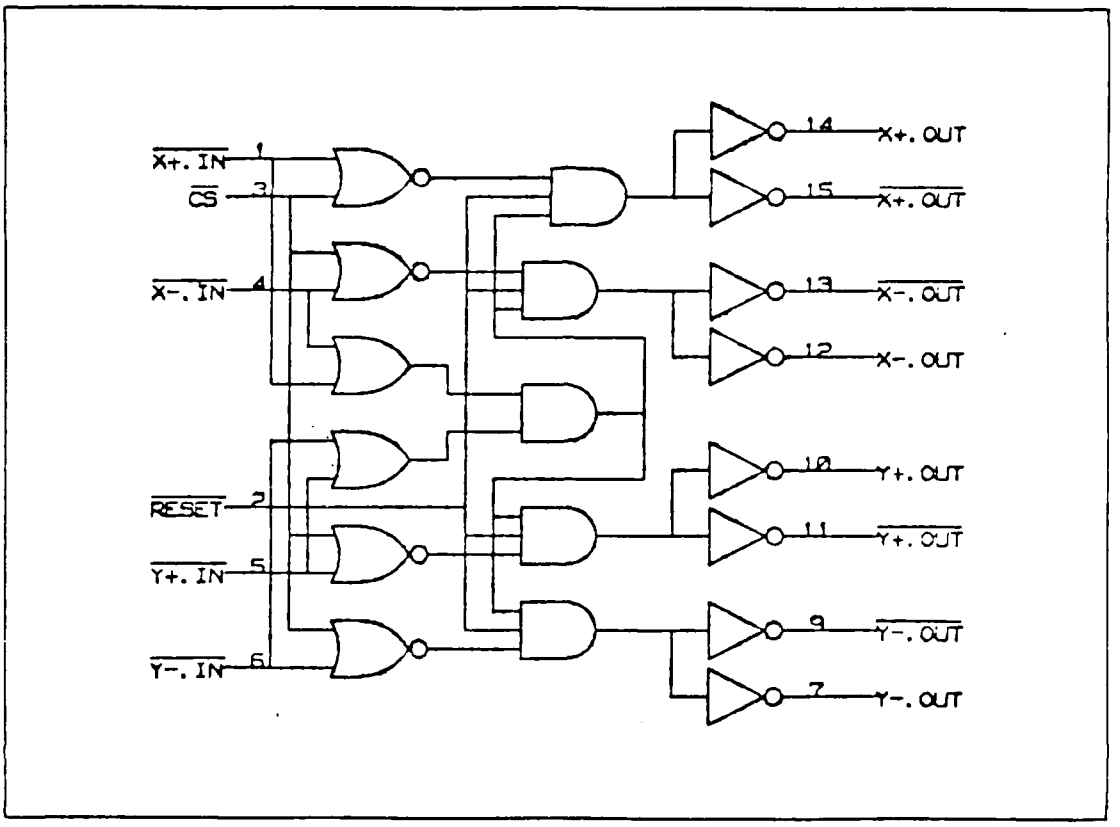


Figure 4.2 CPD Logic Diagram.

The heart of the system is the MBM Controller (BMC). It is the interface between the memory module and its host. The Controller provides all required timing signals. It converts the serial data from the FSA FIFO to parallel data and, conversely, changes parallel data to serial. Figure 4.3 shows the ten functional blocks of the Controller. A brief discussion is given for each of these functional blocks. The reader is referred to [Ref. 9] for a more detailed account.

The Power Fail and Reset (Block 1) is self-explanatory. When activated, the Controller resets the bubble system in an orderly manner.

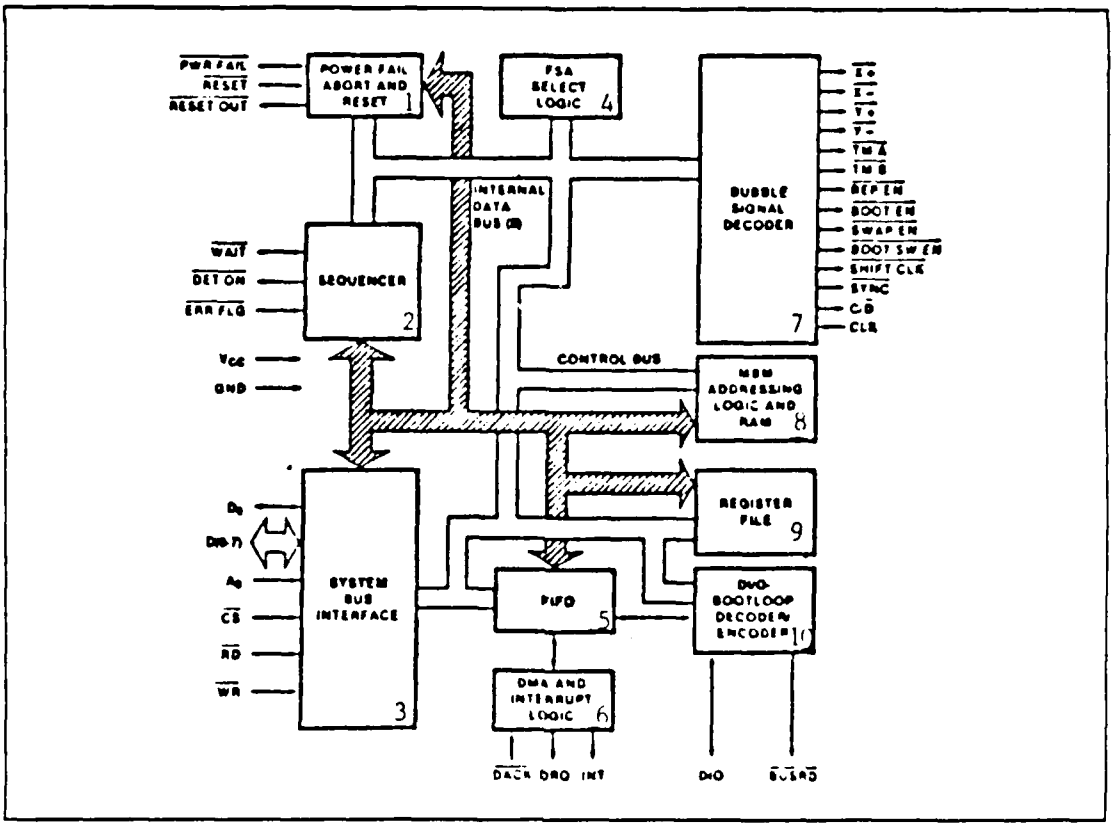


Figure 4.3 Ten Functional Blocks of the Controller.

The Sequencer (Block 2) encompasses the WAIT, DET.ON, and ERR.FLG commands. A step-by-step procedure must be followed by the BMC when one of these commands is given. Within the Sequencer is an internal ROM that contains the pre-programmed instructions used to implement the various commands. Once a command is given and decoded, the Sequencer steps through the instructions pertaining to that command [Ref. 11: p. 21].

The System Bus (Block 3) interfaces the Controller with the host. Commands, status information, address, and data are sent via these lines. In order to know what type of function it is to perform, the Controller's register file must be supplied with specific information (Block 9) before any type of data transfer can take place. The transfer of

this information is done on the 8-bit data bus with bit four set to zero. Figure 4.4 lists the six registers that must be addressed. The 4 Mbit Controller, 7224, does not make use of the Utility Register (UR). The Block Length Register (BLR)determines the system page size and the number of pages to be transferred. The Address Register (AR) defines the page on which the transfer of data is to start. The Enable Register (ER) defines the modes under which the transfer will take place, i.e., interrupt, polled, or DMA. In Figure 4.5, note the BLR and AR require two eight-bit codes, one pertaining to the most significant bits, the other to the least significant bits. Eleven of the bits available in the BLR hold the value for the number of pages to be transferred and provide the user with the possibility of transferring from 1 to 2048 pages. If more than one MBM is connected in parallel, the width of a page can be increased, i.e., 64, 128, 256, 512, etc. The four most significant bit positions hold this information. The four most significant bits of the AR are used in conjunction with the BLR to control the serial selection of bubble memories or a group of memories. The remaining eleven define on which page the transfer is to start [Ref. 7: pp. 7-10].

When data bit four is a one, the information is decoded as a command. There are sixteen commands used by the BMC, (See Figure 4.6) The four most common commands are ABORT, INITIALIZE, READ BUBBLE DATA, and WRITE BUBBLE DATA. Those commands pertaining to the bootloop are used only for diagnostic purposes. The remaining commands provide other options available to the user and are described in [Ref. 9].

Information about data manipulation, such as page size, mode of operation, pages being transferred, etc., is stored in the parametric registers. The A0 line is held high during the programming of these registers. Once all of the required information has been passed, the A0 line goes low and the BMC is ready to transfer data.

Register Name	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Read/ Write
Utility Register	0	0	0	0	1	0	1	0	R/W
Block Length Register (LSB)	0	0	0	0	1	0	1	1	W
Block Length Register (MSB)	0	0	0	0	1	1	0	0	W
Enable Register	0	0	0	0	1	1	0	1	W
Address Register (LSB)	0	0	0	0	1	1	1	0	R/W
Address Register (MSB)	0**	0	0	0	1	1	1	1	R/W
7220 FIFO	0	0	0	0	0	0	0	0	R/W

NOTES:

*With BMC A0 signal = 1

Figure 4.4 Six Parametric Registers.

Block 4, FSA Select Logic, contains the logic that controls all communication between the BMC and the FSA. As mentioned, the serial FIFO of the FSA receives information from and sends information to the FIFO of the BMC. The timing of this transfer is an important factor and is controlled internally by the FSA Logic Block.

The FIFO (Block 5) has the important function of settling timing differences between both the host interface and the BMC and between the FSA and the BMC. The FIFO is dual ported, i.e., it can be written into and read from simultaneously. The maximum amount of information it can hold at any one time is 40 bytes. While the BMC is executing a command, it functions as a data buffer. However, when the BMC has completed transfer of all commands, the FIFO performs as a general-purpose FIFO. As shown in Figure 4.4, the FIFO is automatically addressed after the last of the six parametric registers has been

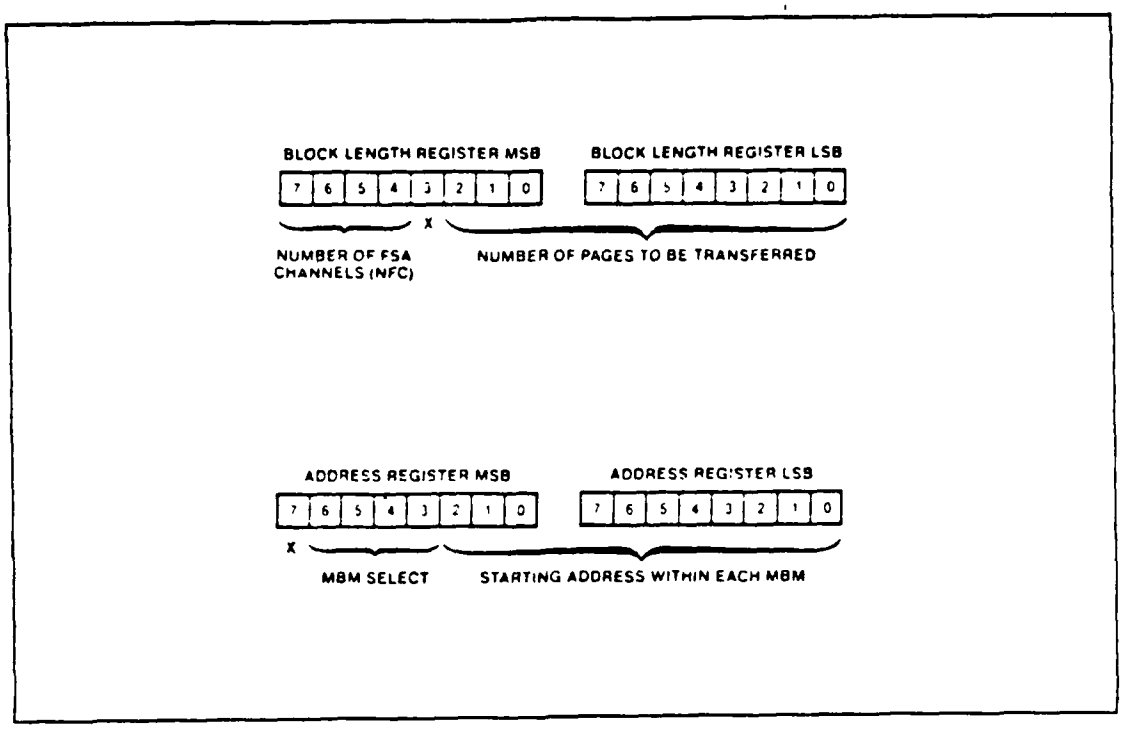


Figure 4.5 Parametric Register Organization.

written to, thereby signalling that the BMC is ready for a data transfer.

Block 6 is concerned with the type of transfer to be used in recording data. The MBM can operate under three different modes:

1. Polled,
2. Interrupt driven,
3. Direct Memory Access (DMA).

Polled Transfer is the easiest mode to implement with the MBM. It is, however, the most time consuming in relation to microprocessor overhead. Interrupt driven transfer requires less microprocessor overhead. This mode also permits transfer of data in blocks of information. The DMA mode is the one that will be used in the NPS GAS experiment. There is no microprocessor overhead in a DMA transfer, and it is the fastest mode in transferring data to the MBM. The

Four-Bit Command Code				Command
D3	D2	D1	D0	
0	0	0	0	Write Bootloop Register Masked
0	0	0	1	Initialize
0	0	1	0	Read Bubble Data
0	0	1	1	Write Bubble Data
0	1	0	0	Read Seek
0	1	0	1	Read Bootloop Register
0	1	1	0	Write Bootloop Register
0	1	1	1	Write Bootloop
1	0	0	0	Read FSA Status
1	0	0	1	Abort
1	0	1	0	Write Seek
1	0	1	1	Read Bootloop
1	1	0	0	Read Corrected Data
1	1	0	1	Reset FIFO
1	1	1	0	MBM Purge
1	1	1	1	Software Reset

Figure 4.6 Commands Used by Bubble Memory Controller.

BMC operates in a standard two-way handshake protocol, utilizing the DRQ and DACK lines found in Block 6 (see Figure 4.3).

The Bubble Signal Decoder (Block 7) contains all the logic needed for generating all MBM timing signals. It is comprised of a three-stage counter, a decoder, and synchronous latches. The first stage of the counter is a divide-by-four counter. The second stage, a divide-by-twenty counter, produces the field rotation frequency. For example, since BMC requires a 4 MHz clock, the output of the first stage is at 1 MHz, while the output of the second stage is at a 50 kHz rate. Any of the clock edges that occur during one complete cycle can be used to set and reset MBM signal latches. The TM.A and TM.B latches go to the CPG and determine the pulse widths for the generation and replication of bubbles.

Block 8, MBM Addressing Logic and RAM, contains two more user-accessible registers, an adder, and the MBM address RAM. This RAM stores the next-available logical page address for each MBM.

Finally, the DI/O Bootloop Decoder/Encoder (Block 10) performs parallel -to-serial and serial-to -parallel conversions between the FIFO data and the serial bit stream on the DI/O line. The BUSRD signal, also generated here, is a signal used to indicate the direction of the data transfer. The third function of this block contains the circuitry that decodes the bootloop data during a READ BOOTLOOP operation and encodes it during a WRITE BOOTLOOP operation [Ref. 11: pp. 20-21].

As is apparent, interfacing the MBM module to its host is an involved task. Timing is of great importance. Generation of pulses and rotating current fields must be accomplished. And, finally, circuitry is needed to transform digital data to magnetic, and vice versa. The current state of the art in bubble memories solves most of these problems for the user. The greatest obstacle is found in producing correctly coded software.

V. DESIGN OF THE SSDR

Before attempting to design the solid state data recorder for the NPS experiment, a prototype was built using the 1 Mbit MBM device. This made it possible to become familiar with both the hardware and the software of the MBM. The main goal was to prove the concept that the MBM could be used as a viable recording medium. The first section of this chapter briefly describes the design and programming of the 1 Mbit device as a mini recorder. The second section describes the 12 Mbyte data recorder used in the GAS experiment. The third section addresses the limitation and possible alternatives to this design.

A. PROTOTYPE

Before incorporating an MBM into a system it is necessary to choose the mode of operation, (i.e., Polled, Interrupt driven, or Direct Memory Access), the level of complexity, and flexibility required for the software drivers.

Since the prototype's function was to sample a sine wave of 1000 Hz frequency or less from only one channel, the Polled Mode (PM) was chosen. Sampling at the Nyquist frequency and using only one channel, provided ample time to perform all the operations needed when using the MBM. (Note: By increasing any part of the system, i.e., bubble memories, channels, etc., a faster mode would have to be chosen, which would result in improved performance.)

The complexity of the software was kept to a minimum, therefore flexibility was restricted. The reasons for designing a prototype were threefold:

1. gain familiarity with the MBM and its compontents,
2. prove it could record and play back accurately,

3. increase the design to the size required for the GAS experiment.

The basic operation for the 1 Mbit device is to sample the waveform through an analog-to-digital converter (A/D), store the data in the MBM, upon command, play back recorded waveforms through a digital-to-analog converter (D/A). The major components used to perform the three operations are the Analog Devices' A-D 570, Intel's 1 Mbit 7110 MBM, and National Semiconductor's DAC0800 D-A.

A dual trace oscilloscope was used to compare the accuracy of the recorded waveform to that of the one being played back from the memory. A signal generator provided the signal.

In the Polled Mode, blocks of data can be transferred into the MBM as long as the parametric registers in the BMC have been programmed accordingly. The DRQ line from the BMC signaled the microprocessor that the MBM was ready for a data transfer.

As explained in Chapter 4, the MBM is written to in pages, each page containing 64 bytes (68 bytes without error checking). The total amount of pages recorded at one time is established under software control. The initial draft of this software transferred one page at a time. Two temporary storage areas, each 64 bytes in size, were reserved in RAM. As one area was filling with data samples, the other area was being sent to the bubble. Timing was found to be extremely critical. After the recorder worked successfully for one page, the number of pages was increased until the record process failed. The maximum number of pages successfully transferred was found to be 270 at a sampling rate of 2 kHz.

Power failure is a concern in any system design. Writing the software so that data transfers of one-page are performed, ensures that the minimum amount of data will be lost. Whatever has been stored in the MBM will remain

intact, but whatever has been stored in RAM will be lost. As a consequence, storing a greater number of pages in RAM results in a greater data loss.

Recalling that communication is not carried out directly with the MBM, interfacing it with the host microprocessor, requires that specific procedures be followed before any data transfer can take place. In order for the BMC to prepare itself for any further instructions, it must receive an ABORT command, followed by an INITIALIZE (INIT) command. Once these commands are accomplished, the BMC's parametric registers are loaded in preparation for the upcoming data transfer. The eight-bit command/status port can be polled to determine whether a successful operation has been performed. If the operation has failed, the OP-FAIL bit will be set; if it has succeded, the OP-COMPLETE bit is set.

Initializing the BMC before writing the parametric registers sets the BMC to a known state. This command results in resetting the support components, placing the bubble at page zero, and enabling the error correction. If the start of a record process is designed to start at a page other than page zero, then the desired page address must be reflected in the information supplied to the address register of the parametric registers. The block length register must be programmed with the number of channels to be used and the number of pages to be transferred.

The software program of the prototype made available to the user a menu which displayed a number of options to choose from. These options included:

1. Setting the sample rate,
2. Sampling without recording,
3. Record,
4. Playback,
5. Initialize the bubble.

Upon selection of an option, the program immediately executed it. Option 5 had to always be performed first. Additionally, the same sample rate chosen for RECORD had to be used for PLAYBACK. Option 2 was used primarily to establish the successful operation of the A/D and D/A.

Tests were performed on the 1 Mbit device. These tests included turning off power, changing the frequency of the sampled waveform while recording, changing the waveform itself while recording, and attaching a microphone/speaker arrangement in order to record a signal other than a pure sine wave. In all cases, the MBM performed successfully. It did not lose data when power was removed. It followed the changes in frequency and waveform with only a small phase shift due to the time delay. There was, however, distortion in the sound recording. This was due in part to the low sample rate and to the quality of the microphone and speaker being used.

B. SOLID STATE DATA RECORDER (SSDR)

The final design of the SSDR has the capability to store twelve mega-bytes (twenty-four four-megabit prototype MBM cards) of data. The 8085 microprocessor is at the heart of the system. A basic block diagram--a "generic" data recorder--can be found in Figure 5.1. Subsection 1 describes the SSDR as a general purpose data recorder. Subsection 2 describes how this general purpose recorder was adapted to the NPS GAS experiment.

As in the prototype model, the desired information is sampled by the A/D device. The digitized information is then temporarily transferred to a RAM buffer. Upon request, the DMA will gain control of the bus and perform the data transfer from the buffer to the MBM where it will be stored. The ideal data transfer rate of the four-megabit (4 Mbit) MBM is 25 Kbytes/sec [Ref. 7: pp. 3-4]. Due to an internal delay known as T-SEEK, the maximum obtainable data transfer

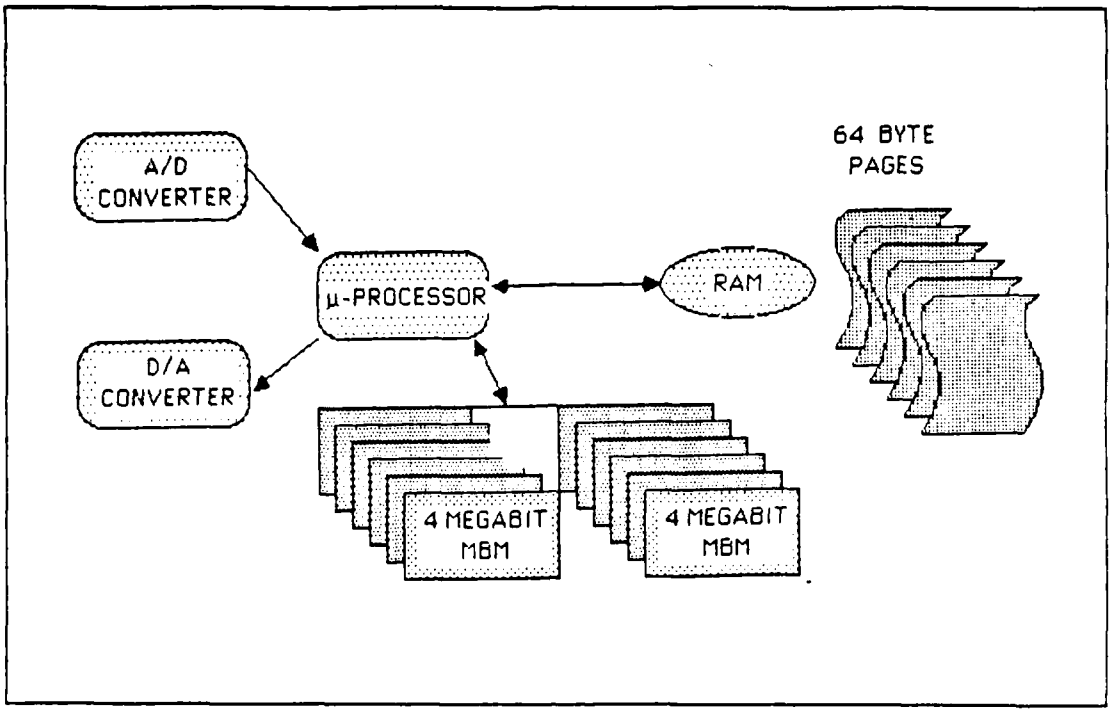


Figure 5.1 Generic Data Recorder.

rate varies. T-SEEK is defined as the amount of time between the issuing of the READ (or WRITE) command to the MBM and the MBM locating the page to be read (or written). T-SEEK can range in values from 20 microseconds to 163,820 microseconds with an average of 81,920 microseconds.

By disguising T-SEEK so that the SSDR does not "feel" its effect, i.e., have an inherent delay everytime a READ or a WRITE occurs, and by using a buffer 32 Kbytes in size, the data transfer rate of 17 Kbytes/sec is obtained. T-SEEK is disguised by using a hardware comparator. Calculations were performed to find at what RAM buffer location a WRITE command needed to be issued, so that, by the time a full storage area of the buffer was filled, the MBM would be ready for the data transfer. A comparator was then connected having the same address as that location. Upon addressing this location, the comparator generates the

appropriate interrupt, signaling the 8085 to issue the WRITE command (see Figure 5.2). If it were feasible to increase the buffer to store a full 4 Mbits of data, and by the additional elimination of T-SEEK, the ideal data transfer rate of 25 Kbytes/sec could be achieved.

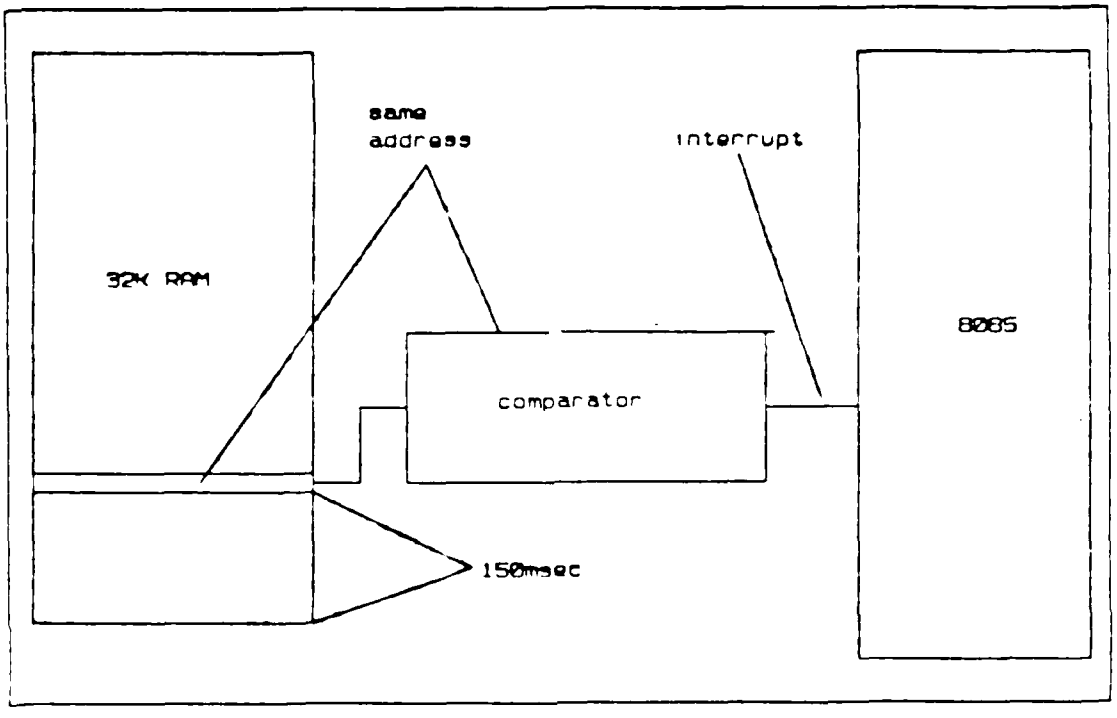


Figure 5.2 Implementation of Hardware Comparator.

1. General Purpose SSDR

The SSDR has the following characteristics. The sample rate is fixed at 2.5 KHz. As a result, (and due to the T-SEEK/buffer limitations) the maximum obtainable data rate for one eight-bit channel is 17 Kbytes/sec.

The MBMs are adaptable to a power switching configuration, therefore the active MBM card need only be the one being written to or read from. Power consumption is kept to a minimum by the use of a custom designed power switching card, the "ECARD," (see Appendix A.) Thus, power is 15 Watts total for this design. The maximum storage capacity

is 12 Mbytes. Baseline record time (i.e. one channel, 8-bit resolution) is 83.9 minutes continuous recording.

2. Adaptation of the SSDR to the NPS GAS Experiment

The following is a description of how the general purpose SSDR is configured to fulfill the requirements of the NPS experiment for Shuttle.

Six eight-bit channels are required to obtain all the data. An in-depth study was performed by students working on the acoustic section of the experiment to best determine how to implement these channels and is not discussed in this thesis. The results are that three microphones and two accelerometers are connected to five A/D's. The A/D's have up to sixteen data lines from which to obtain information. Since the SSDR has only eight-bit resolution, the acoustic group also determined which of the sixteen lines would be connected to each eight-bit channel. The A/D's are then strobed synchronously to prevent any time delays encountered when analyzing the data. Once strobed, the channels are read using time division multiplexing (TDM), always reading the channels in numerical order, (see Figure 5.3). The maximum obtainable data rate at 2.5 KHz for the six channels is 15 Kbytes/sec.

The full 12 Mbytes of storage will be utilized. The organization of the memory is accomplished in the following manner under software control. (Appendix B provides the software code used to run the SSDR.) Three record options are available to chose from and they are SWEEP, SCROLL, and LAUNCH. SWEEP is to be performed prior to launch. A tone will be generated by a voltage controlled oscillator (VCO) starting at 25 Hz and stepping up to 1000 Hz at 1 Hz intervals. This will be done in order to excite and record the fundamental acoustic modes in the STS cargo bay. Seventeen of the twenty-four MBM cards are required to store this data and will take 16.5 minutes. The recorder will then go into a standby mode awaiting its next command.

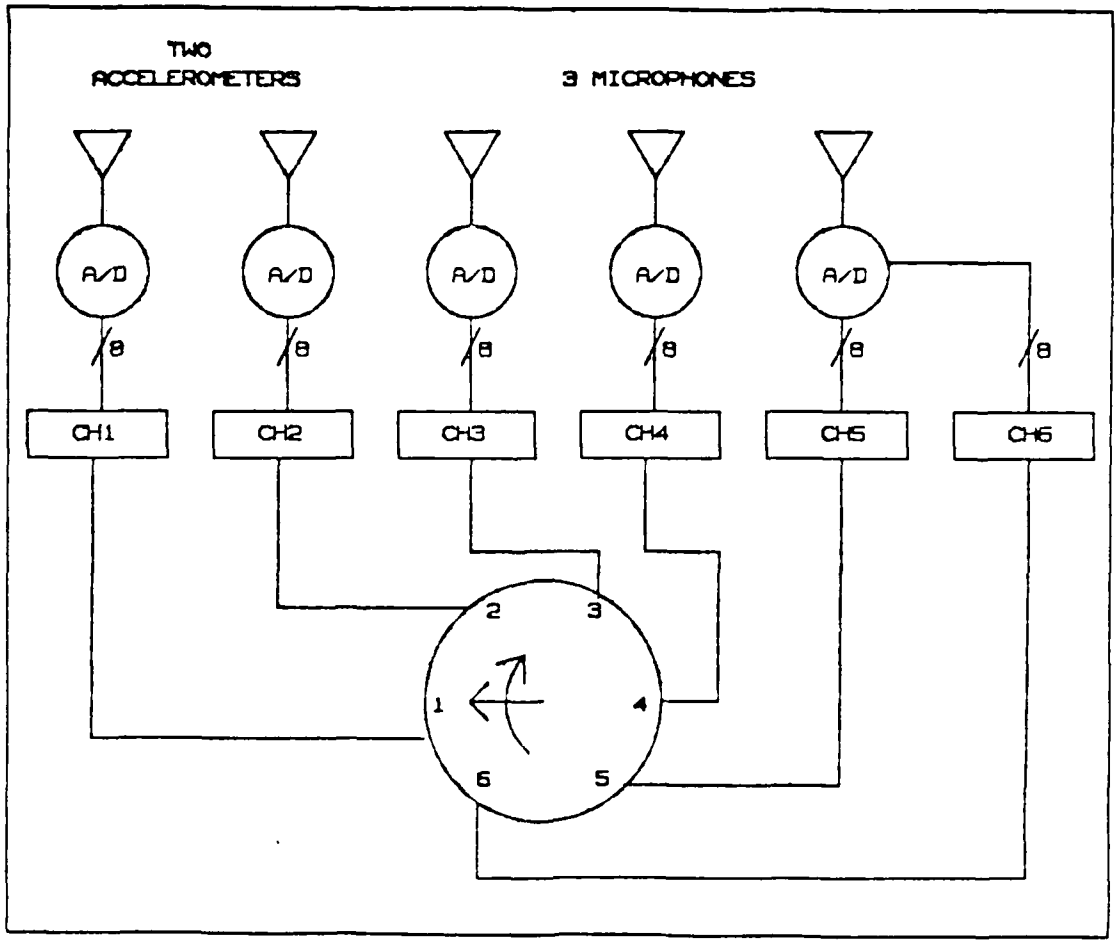


Figure 5.3 Basic Channel Configuration.

SCROLL, the next record option, will be initiated by the powering up of the Auxiliary Power Unit (APU) on the Shuttle. This initiation process is done by the use of a matched filter and is the topic of a thesis written by LT D. W. Jordan, USN, titled A Matched Filter Algorithm for Acoustic Signal Detection. During SCROLL, two MBM cards will be continuously recorded on until launch. This option provides the capability to capture the very important information otherwise lost should the record process be initiated by a detection of the launch itself.

Once launch is detected, however, the recorder will transition immediately to the first of the remaining six MBM cards and record the information to be obtained before the Shuttle leaves the earth's atmosphere. At this point, the job of the data recorder is complete and it will be shut down, retaining the data for analysis when it returns to NPS. The total record time for the SSDR is 21 minutes. It should be noted, however, these options do not have to follow consecutively, i.e. should the matched filter not function properly and a launch is detected, the data recorder will jump to the specified MBM card and record the remainder of the launch. The determination of a launch is not done by the SSDR and is not a topic discussed in this thesis. For more information concerning the control of the experiment as a whole, the reader is referred to a thesis written by Lt J. W. Wallin, USN titled Microprocessor Controller with Nonvolatile Memory Implementation.

3. Limitations and Alternative

Some of the limitations of this design are a result of the prototype MBM card used in the experiment. For example, the 43 byte FIFO in the BMC limits the data transfer rate. One reason the FIFO exists is to smooth out timing differences between the host and the MBM. Only having 43 bytes for the buffer has proved to be inadequate. INTEL has since built the 4 Mbit device with a 128 byte FIFO.

The performance of a particular bubble memory device ultimately is a function of three factors,

1. the number of storage loops,
2. the number of stoarge locations on those storage loops,
3. the frequency of the rotating magnetic field (coil frequency).

The throughput, which is number of bytes/sec, is directly proportional to the number of storage loops and the coil

frequency. The time to find a particular page of data (T-SEEK) is directly proportional to the number of storage locations and inversely proportional to the coil frequency. Power is directly proportional to the coil frequency. Increasing or decreasing any of the three changes the performance and trade-offs must be made.

The 8085 microprocessor is an 8-bit device. Designing the SSDR to perform with a 16-bit device or a 32-bit device, would enhance the resolution and provide for data throughput of up to 272,000 Mbytes/second. For more information on a 32-bit design, the reader is referred to Lt T. J. Frey's thesis, A 32-bit Microprocessor Based Solid State Data Recorder for Space-based Applications.

The SSDR is limited to 12 Mbytes of storage. This is partly due to size, weight, and power constraints of the GAS container. It is also limited to this size because of the use of the prototype MBM card. Using "off the shelf" components benefitted the GAS experiment as far as time and money was concerned, however, flexibility was affected.

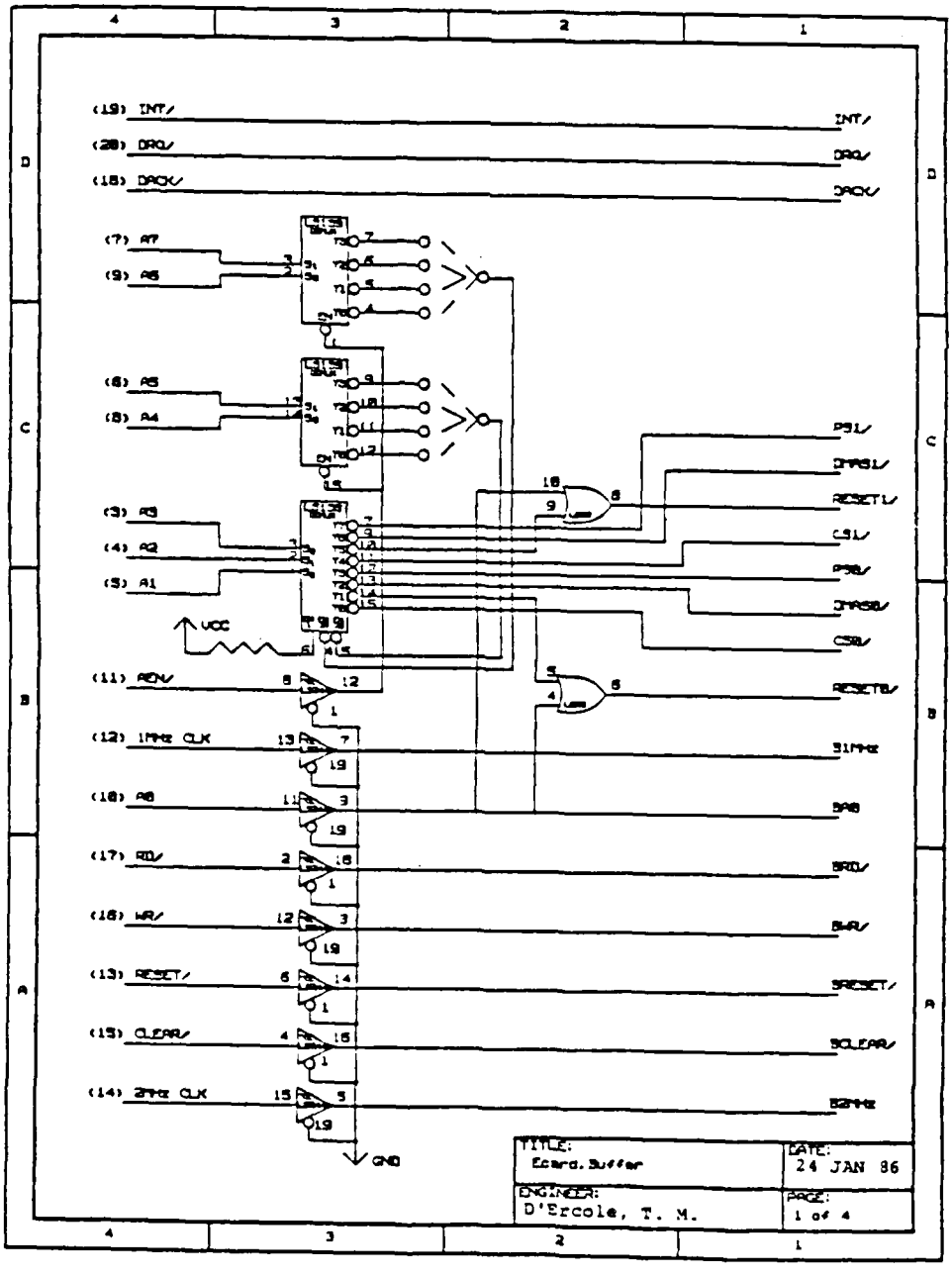
An alternative, not only for the MBM card but for the SSDR as a whole, would be to custom design a card enabling the BMC to control more than one MBM. The BMC can control up to eight MBMs at one time [Ref. 9]. For further information the reader is referred to Lt. B. A. Campbell's thesis, A Digital Recording System for Space-based Applications Utilizing Four-megabit Magnetic Bubble Memories.

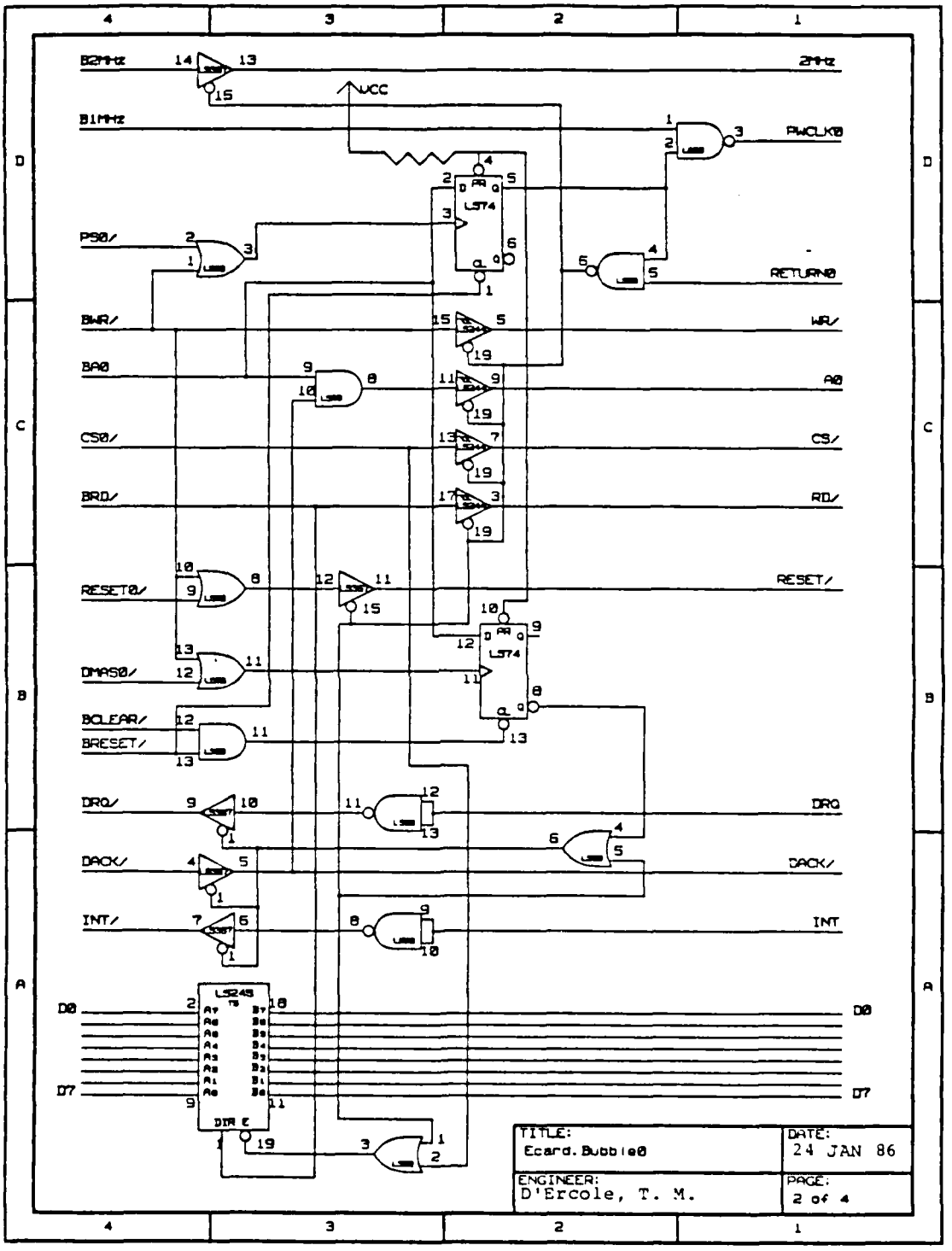
The NPS GAS experiment is scheduled to fly in 1986. Analysis of the data and performance of the SSDR is another phase of the experiment as a whole. The scope of this thesis concentrates solely on the MBM and its implementation into a data recorder. Because of the MBM's inherent hardness to radiation, ruggedness and reliability, its future use in space-related or space-born systems should not be underestimated.

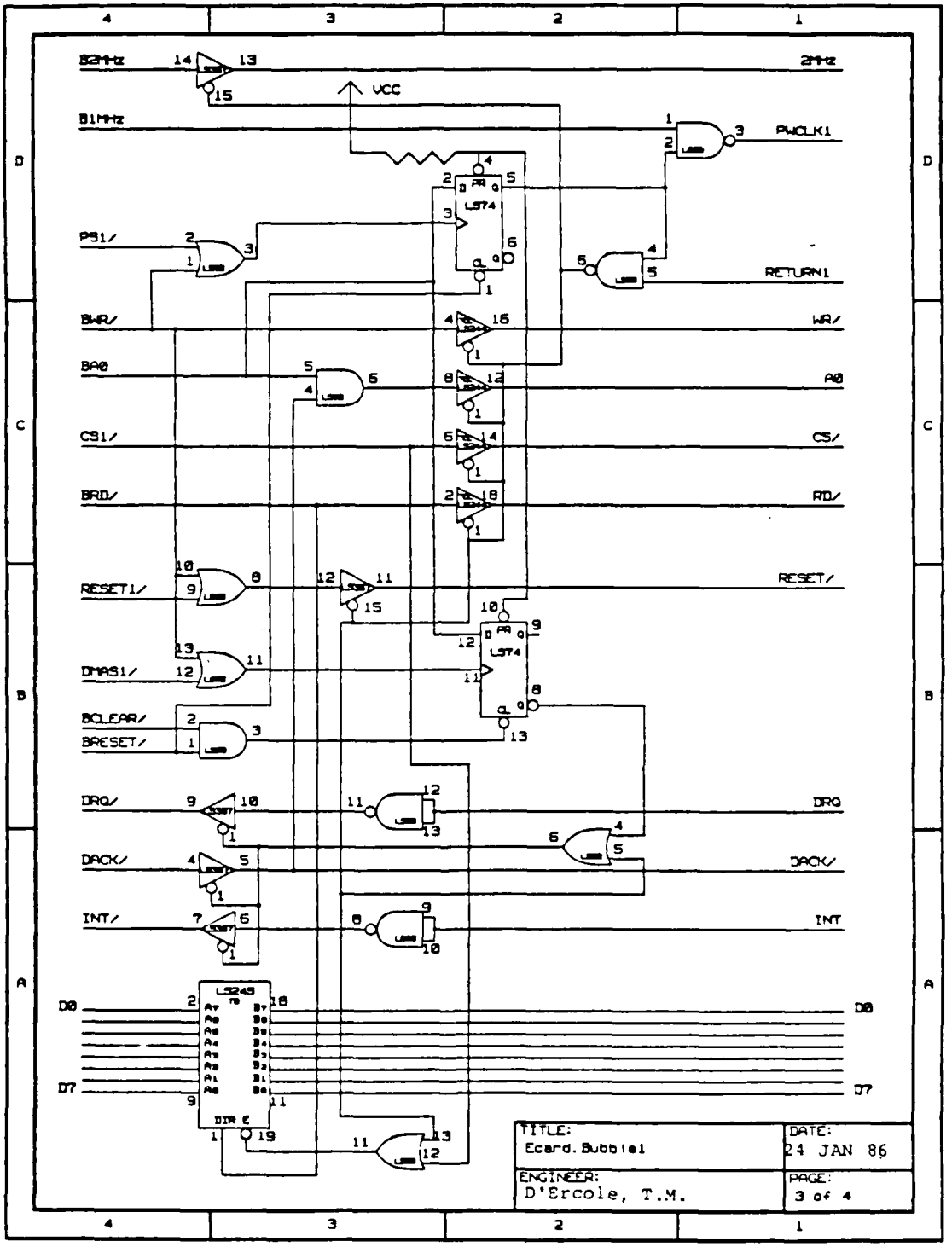
APPENDIX A

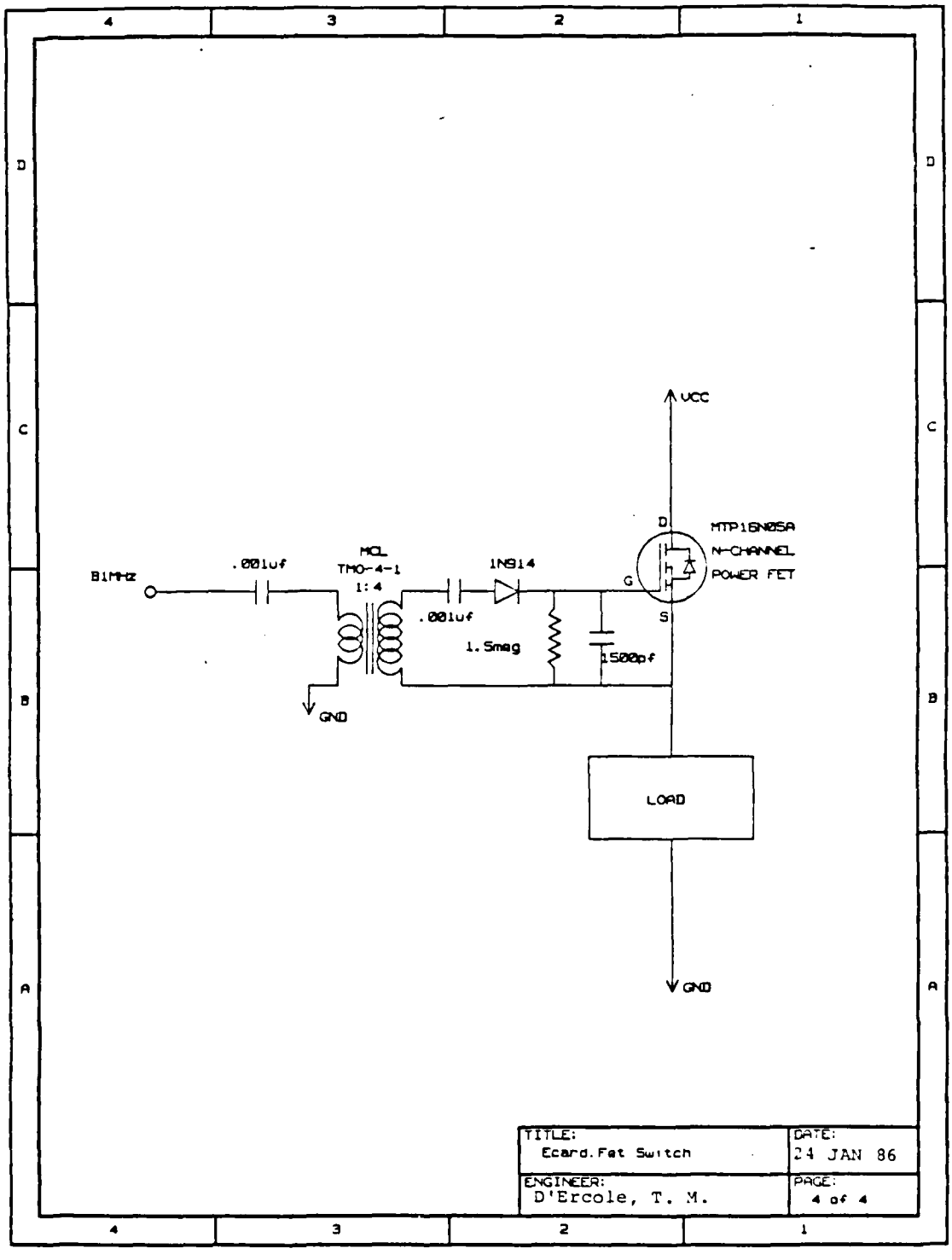
SCHEMATIC OF THE SOLID STATE RECORDER

A. POWER SWITCHING CARD (ECARD)

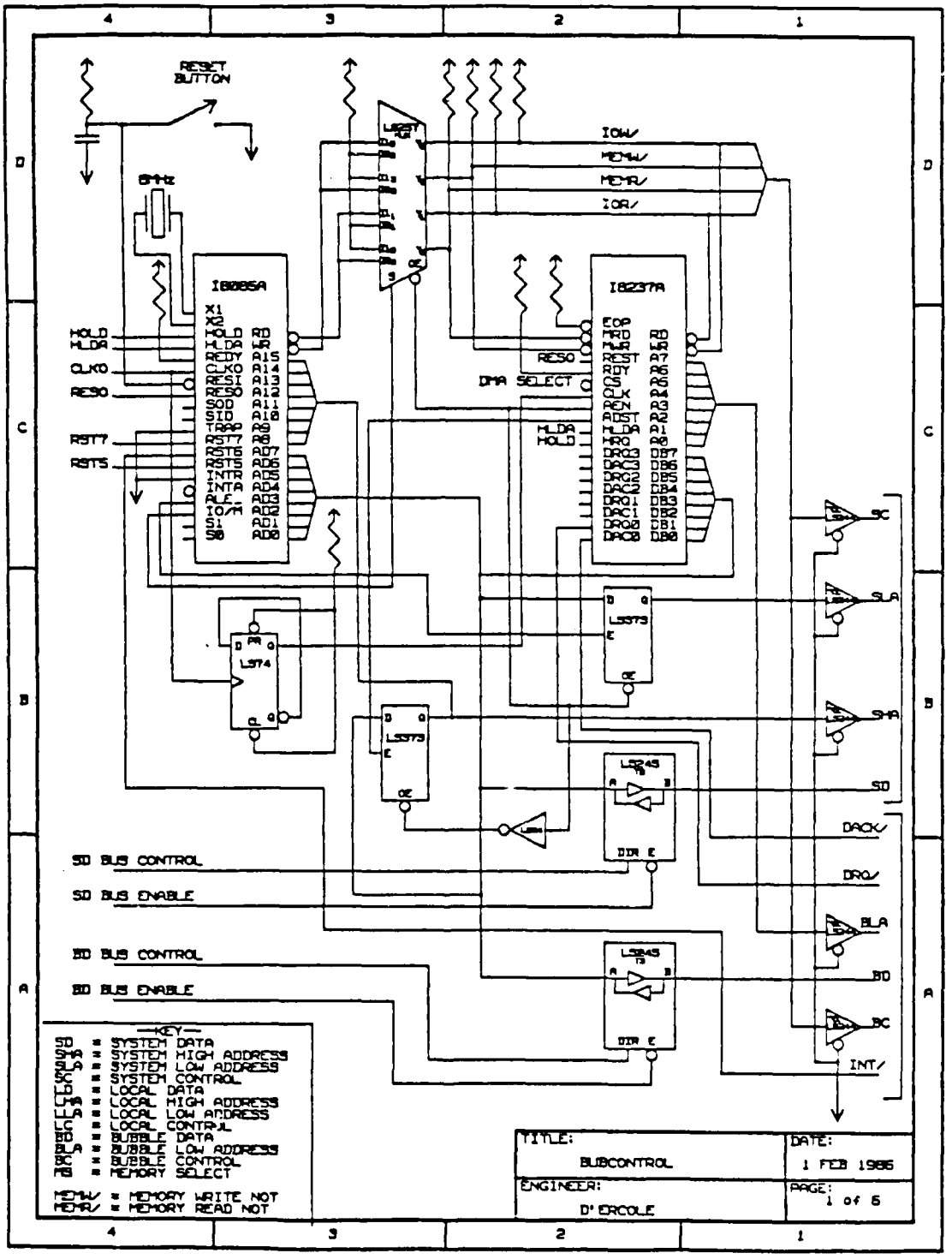


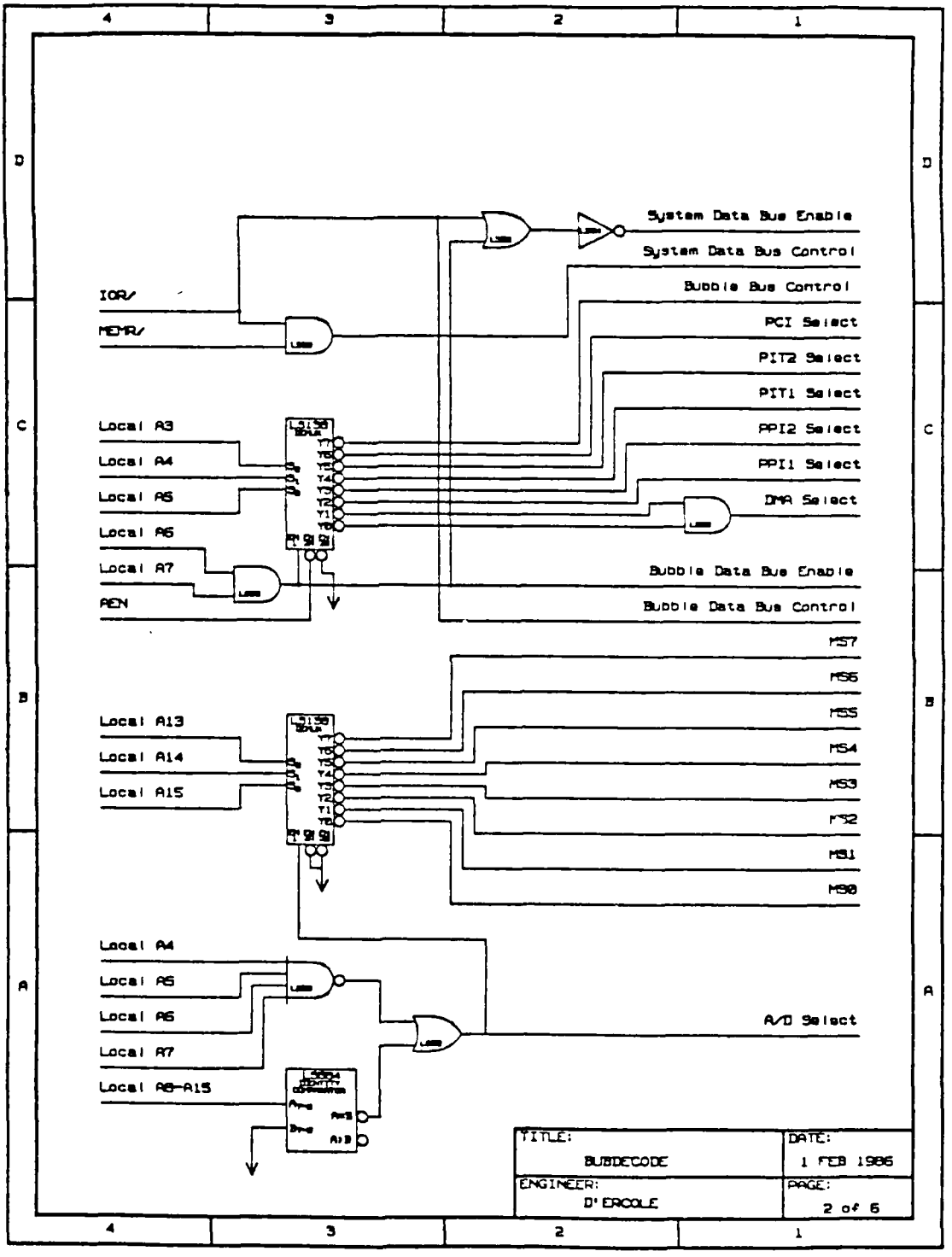

B2MHz
B1MHz
VCC
2MHz
PWCLK0
PS0/
RETURN0
BWR/
WR/
BA0
A0
CS0/
CS/
BRD/
RD/
RESET0/
RESET/
DMAS0/
BCLEAR/
BRESET/
DRQ/
DRQ
DACK/
DACK/
INT/
INT
D0
D7
LS245
LS74
TITLE:
Ecard.Bubble0
DATE:
24 JAN 86
ENGINEER:
D'Ercole, T. M.
PAGE:
2 of 4

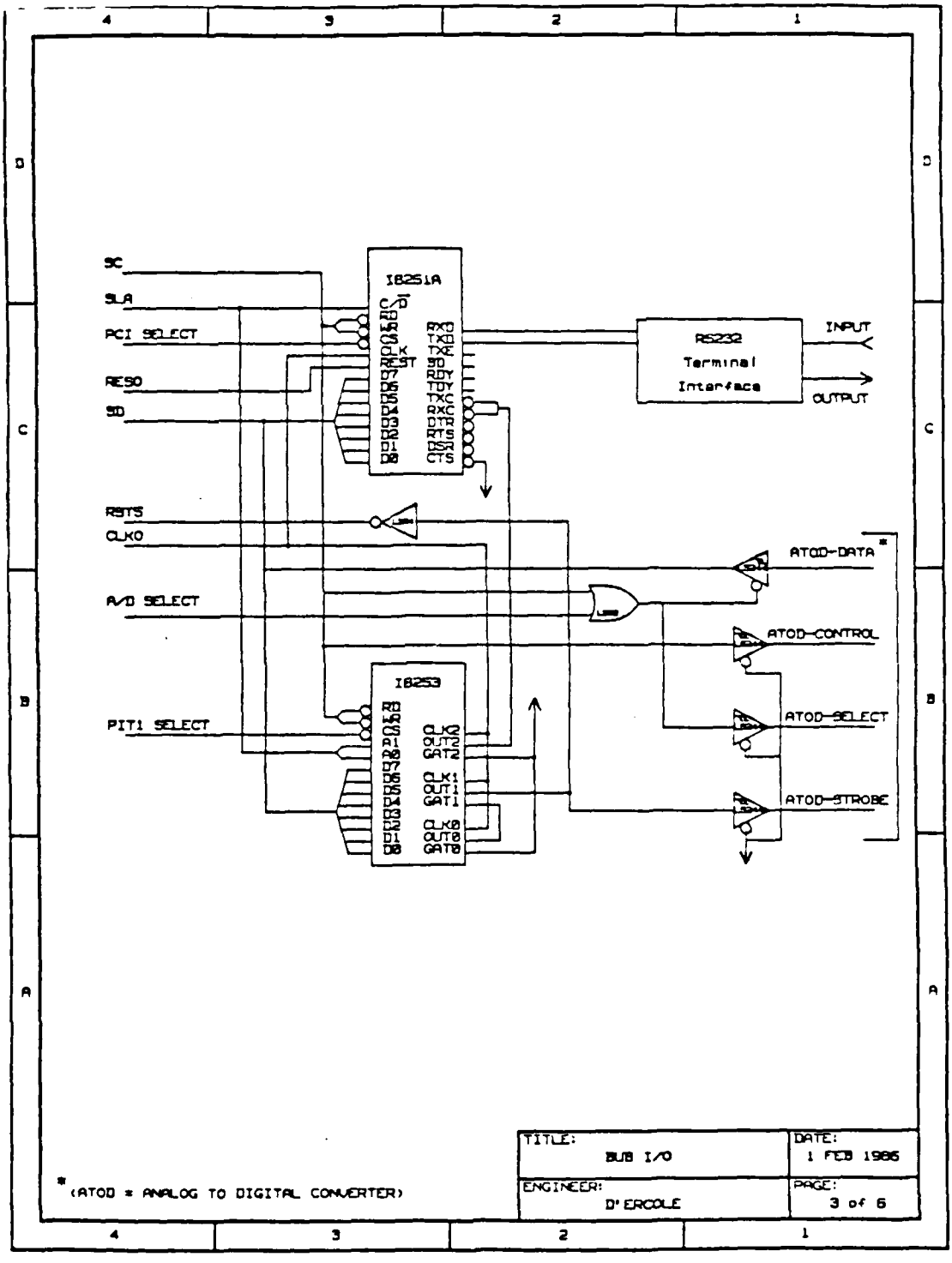

4
3
2
1
D
C
B
A
8214Hz
2Hz
VCC
8.1MHz
PWCLK1
PS1/
RETURN1
BWR/
WR/
BA0
A0
CS1/
CS/
BRD/
RD/
RESET1/
RESET/
DMAS1/
BCLEAR/
BRESET/
DRQ/
DRQ
DACK/
DACK/
INT/
INT
D0
D7
LS374
LS245
TITLE:
Ecard.Bubble1
DATE:
24 JAN 86
ENGINEER:
D'Ercole, T.M.
PAGE:
3 of 4

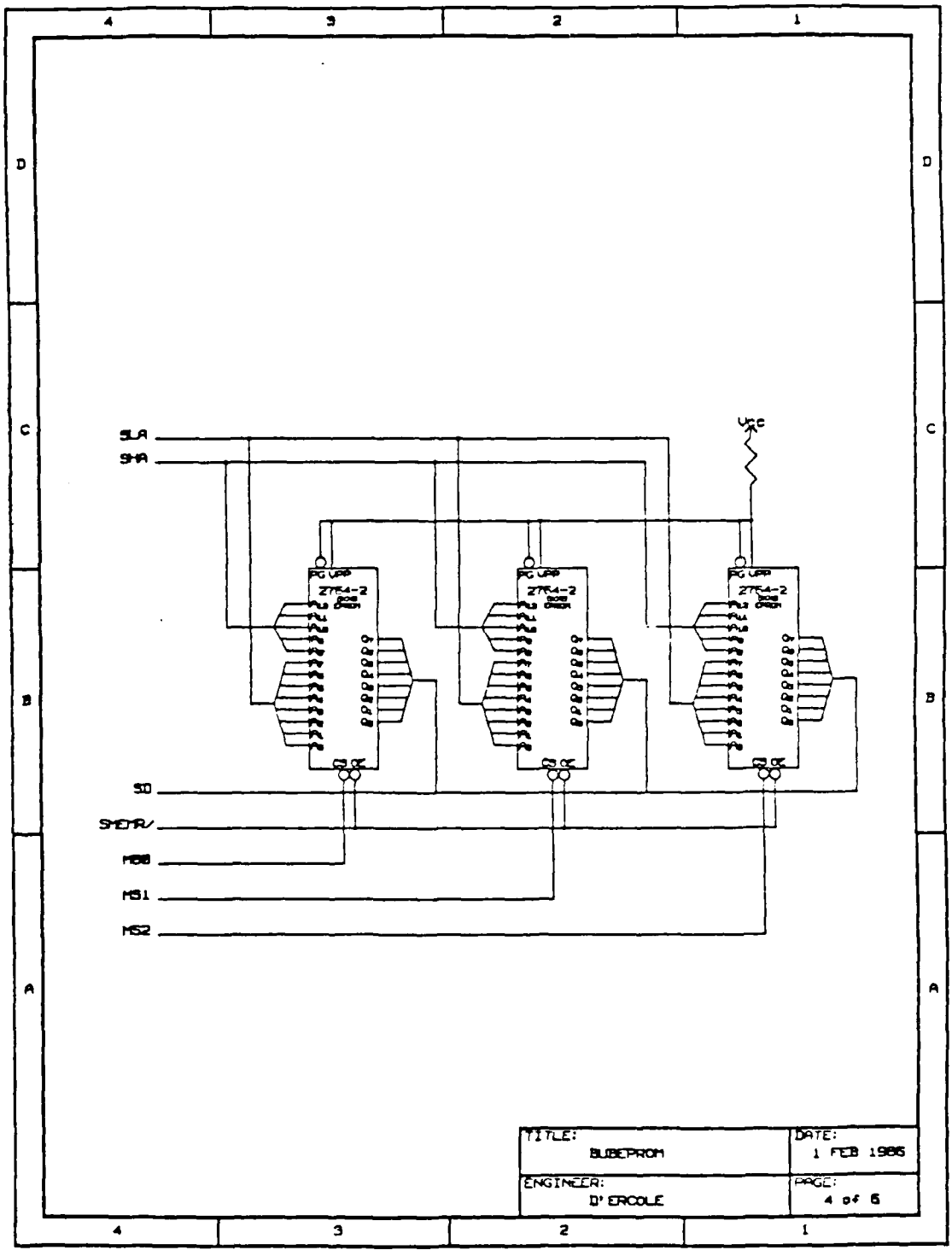

4
3
2
1
D
C
B
A
VCC
D
MTP16N05A
N-CHANNEL
POWER FET
G
S
81MHz
.001uf
MCL
TMO-4-1
1:4
1N914
.001uf
1.5meg
1500pf
GND
LOAD
GND
TITLE:
Ecard.Fet Switch
DATE:
24 JAN 86
ENGINEER:
D'Ercole, T. M.
PAGE:
4 of 4

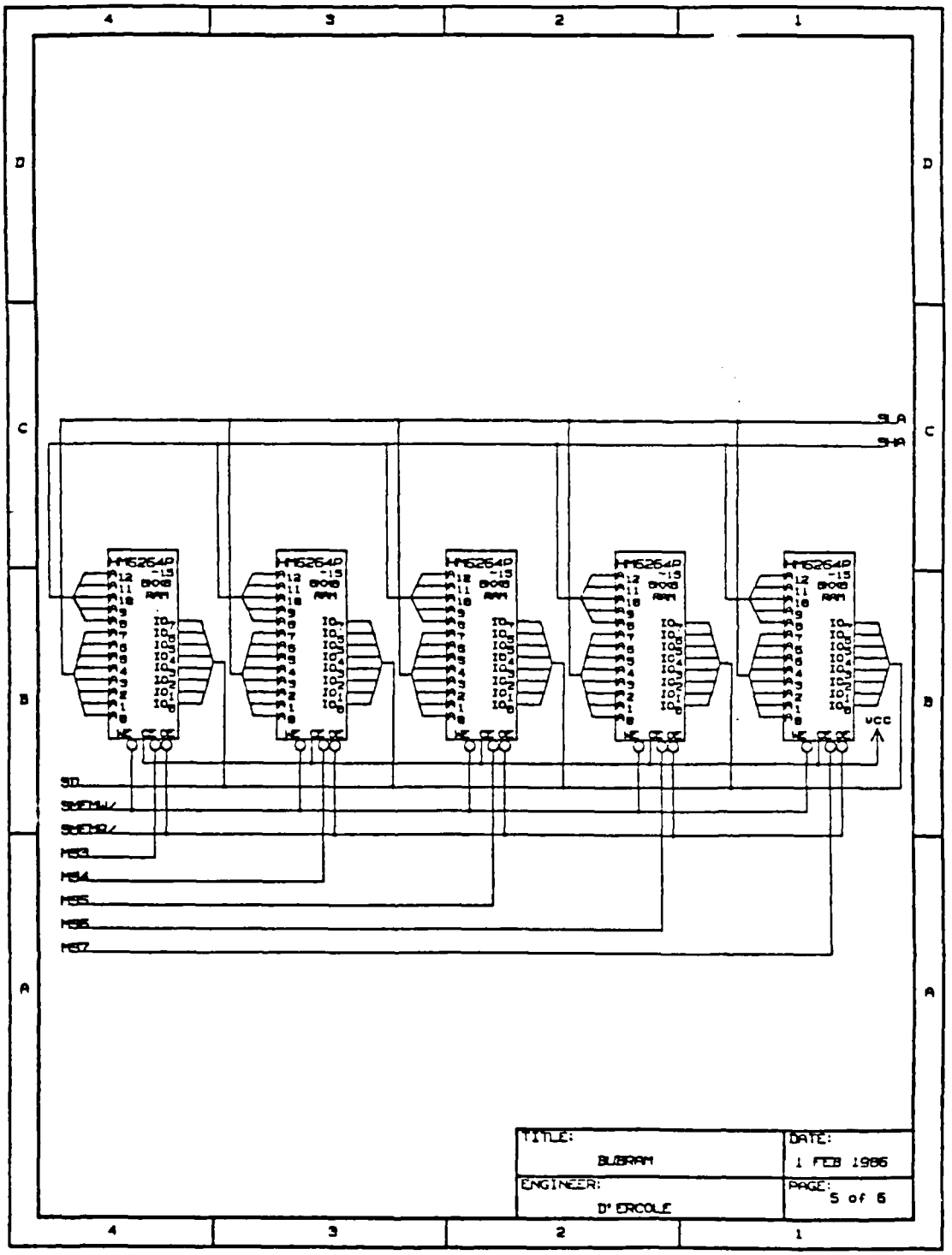
B. CONTROLLER CARD (MICROPROCESSOR/DMA)

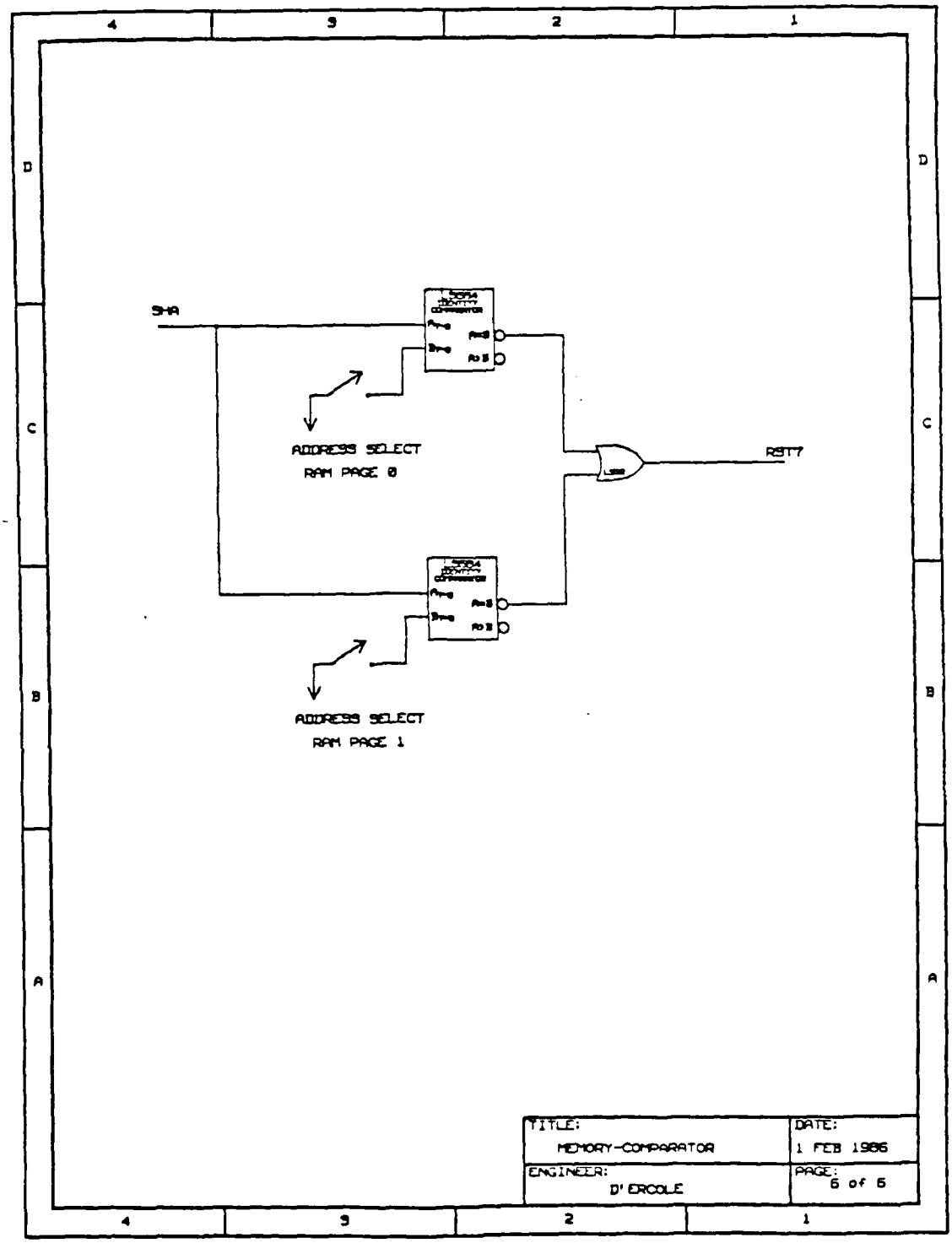



System Data Bus Enable
System Data Bus Control
Bubble Bus Control
PCI Select
PIT2 Select
PIT1 Select
PPI2 Select
PPI1 Select
DMA Select
Bubble Data Bus Enable
Bubble Data Bus Control
IOR/
MEMR/
Local A3
Local A4
Local A5
Local A6
Local A7
AEN
MS7
MS6
MS5
MS4
MS3
MS2
MS1
MS0
Local A13
Local A14
Local A15
Local A4
Local A5
Local A6
Local A7
Local A8-A15
A/D Select
TITLE:
BUBDECODE
DATE:
1 FEB 1986
ENGINEER:
D'ERCOLE
PAGE:
2 of 6


SC
SLA
PCI SELECT
RESO
SD
RSTS
CLKO
A/D SELECT
PITI SELECT
I8251A
I8253
RS232
Terminal
Interface
INPUT
OUTPUT
ATOD-DATA *
ATOD-CONTROL
ATOD-SELECT
ATOD-STROBE
* (ATOD = ANALOG TO DIGITAL CONVERTER)
TITLE: BUB I/O
DATE: 1 FEB 1986
ENGINEER: D'ERCOLE
PAGE: 3 of 6


4
3
2
1
D
C
B
A
SLA
SHA
SD
SHDATA/
MS0
MS1
MS2
TITLE:
BUBEPROM
DATE:
1 FEB 1986
ENGINEER:
D'ERCOLE
PAGE:
4 of 6


SLA
SHA
HM6264P
-15
8K8
RAM
VCC
SD
SMEMW/
SMEMR/
MS3
MS4
MS5
MS6
MS7
TITLE:
BUBRAM
DATE:
1 FEB 1986
ENGINEER:
D'ERCOLE
PAGE:
5 of 6


4
3
2
1
D
C
B
A
SHA
ADDRESS SELECT
RAM PAGE 0
RST7
ADDRESS SELECT
RAM PAGE 1
TITLE:
MEMORY-COMPARATOR
DATE:
1 FEB 1986
ENGINEER:
D'ERCOLE
PAGE:
6 of 6

APPENDIX B

SOFTWARE CODE

```
        EXTRN   INBUEL,RDBUEL,WREUBL,BICTMP
;
        ENTRY   BUBNUM,RDDTMP,WRDTMP,RDSTMP,WRCTMP
;
        CSEG
;
;       SOLID STATE DATA RECORDER PROGRAM
;       FOR 4-MEGABIT BUBBLE MEMORY BOARD
;
STACK   EQU     0               ;STACK POINTER LOCATION
;
TPPICS  EQU     10H             ;TIMER CONTROL/STATUS PORT
TPPIPA  EQU     11H             ;BUBBLE POWER PORT 8255
TPPIPB  EQU     12H             ;BUBBLE SELECT PORT 8255
TPPIPC  EQU     13H             ;NOT USED
TLSB    EQU     14H             ;TIMER LEAST SIG. BYTE 8155
TMSBYM  EQU     15H             ;TIMER MOST SIG. BYTE 8155
;
PPIPA   EQU     20H             ;PROGRAMABLE PERIPHERAL INTERFACE PORT A
PPIPB   EQU     21H             ;PROGRAMABLE PERIPHERAL INTERFACE PORT B
PPIPC   EQU     22H             ;PROGRAMABLE PERIPHERAL INTERFACE PORT C
PPICS   EQU     23H             ;PROGRAMABLE PERIPHERAL INTERFACE STATUS
;
CONDATA EQU     30H             ;CONSOLE DATA
CONSTAT EQU     31H             ;CONSOLE STATUS
;
ATCD    EQU     0C000H          ;ANALOG TO DIGITAL ADDRESS
DTOA    EQU     0C001H          ;DIGITAL TO ANALOG ADDRESS
;
CR      EQU     0DH             ;CARRIAGE RETURN
LF      EQU     0AH             ;LINE FEED
BS      EQU     08H             ;BACK SPACE
;
TXCNT   EQU     100H            ;NUMBER OF BLOCK TRANSFERS
;
PAGE0   EQU     0E000H          ;RAM AREA 0 (2K X 64 BYTES)
PAGE1   EQU     0E800H          ;RAM AREA 1 (2K X 64 BYTES)
;
;TABLE OF RELATIVE ADDRESSES OF VARIABLES AND JUMP VECTORS
;
RAM     EQU     0F000H          ;BEGINNING ADDRESS OF RAM
RAMTABL EQU     RAM             ;BEGINNING OF RAM TABLE
ATODTMP EQU     RAMTABL+100H    ;A TO D TEMPORARY STORAGE ADDRESS
DTCATMP EQU     ATODTMP+2       ;D TO A TEMPORARY STORAGE ADDRESS
COUNT   EQU     DTCATMP+2       ;# OF 64 BYTE PAGES TRANSFERED AS A BLOCK
BUBNUM  EQU     COUNT+2         ;CURRENT BUBBLE BEING USED STORED HERE
RDITMP  EQU     BUBNUM+2        ;JUMP VECTOR CMD TO READ CURRENT BUBBLE
WRITMP  EQU     RDITMP+3        ;JP VECTOR CMD TO WRITE TO CURRENT BUBBLE
RDSTMP  EQU     WRDTMP+3        ;JP VECTOR CMD TO READ STATUS CURRENT BUBBLE
WRCTMP  EQU     RDSTMP+3        ;JP VECTOR CMD TO WRITE COMMAND BUBBLE REG
INTVEC  EQU     WRCTMP+3        ;7.5 INTERRUPT VECTOR CMD
RAMADD  EQU     INTVEC+3        ;CURRENT RAM ADDRESS TO READ OR WRITE FROM
```

```
PAGFUL  EQU     RAMADD+2        ;FLAG BYTE FOR PAGE EITHER EMPTY OR FULL
LEDNUM  EQU     PAGFUL+1        ;FLAG BYTE FOR LED EITHER ON OR OFF
DMANUM  EQU     LEDNUM+1        ;CURRENT BUBBLE DMA IS WORKING W/ STORED HERE
DOCHK   EQU     DMANUM+1        ;INT VECTOR BUBBLE SERVICE ROUTINE
BUBCARD EQU     DOCHK+1         ;MAX # OF BUB CARS AVAILABLE TO SYSTEM
;
;SOFTWARE INTERRUPTS
;
BOOT:   JMP     SYSTEM          ;JUMP TO START OF PROGRAM
        DS      5
RSTRT1: JMP     INBUBL          ;JUMP TO INITIALIZE BUBBLE
        DS      5
RSTRT2: JMP     RDBUBL          ;JUMP TO READ BUBBLE
        DS      5
RSTRT3: JMP     WRBUBL          ;JUMP TO WRITE BUBBLE
        DS      5
RSTRT4: JMP     DO5             ;JUMP TO DATA ERROR TEST
        DS      1
PWFAIL: JMP     DOTRAP          ;JUMP TO POWER FAILURE ROUTINE
        DS      1
RSTRT5: JMP     DO5             ;JUMP TO DATA ERROR TEST
        DS      1
RSTRT55:JMP     DO55            ;JUMP TO EXTERNAL TRIGGER ROUTINE
        DS      1
RSTRT6: JMP     DO6             ;JUMP TO CLEAR RAM BUFFER
        DS      1
;
;INTERRUPT 6.5 INITIATES THE SERVICE ROUTINE WHICH CHECKS THE CURRENT
;BUBBLE FOR AN ERROR INTERRUPT OR OP-COMPLETE
;
RSTRT65:POP     B               ;SAVE REGISTER B-C
        JMP     DOCHK           ;JUMP TO CHK BUBBLE INTERRUPT OR OP-COMPLETE
RSTRT7: JMP     DO7             ;JUMP TO 7.5 SERVICE ROUTINE
        DS      1
;
;INTERRUPT 7.5 INITIATES THE ANALOG TO DIGITAL OR DIGITAL TO ANALOG
;SERVICE ROUTINE WHICH EITHER SENDS OR RECEIVES A BYTE FROM THE RAM
;
RSTRT75:JMP     INTVEC          ;JUMP TO SERVICE ROUTINE
;
;
SYSTEM:                         ;START OF THIS PROGRAM
        DI
        LXI     SP,STACK
        MVI     A,5
        CALL    DELAY
        CALL    INITHW
        LXI     D,MENU
        CALL    PRINT
;
;THIS ROUTINE INPUTS A CHARACTER FROM THE CONSOLE AND ECHOES IT BACK TO THE
;CONSOLE.  IT DETERMINES WHAT FUNCTION IS TO BE PERFORMED BY POINTING THE
;PROGRAM COUNTER TO THE PROPER JUMP STATEMENT LOCATED AT THE SYSTEM TABLE
;(SYSTBL).  THE PROGRAM UNDER NORMAL OPERATION RETURNS TO THIS ROUTINE.
;
```

```
AGAIN:
        LXI     SP,STACK
        CALL    CONIN
        MOV     C,A
        CALL    CONOUT
        MOV     A,C
        CPI     '0'
        JM      ERROR
        CPI     '8'
        JP      ERROR
        SUI     '0'
        MOV     C,A
        ADD     A
        ADD     C
        MOV     C,A
        XRA     A
        MOV     B,A
        LXI     H,SYSTBL
        DAD     B
        PCHL
;
;SYSTBL IS THE BASE ADDRESS FOR THE JUMP ROUTINES.  JUMPS ARE SELECTED BY THE
;AGAIN ROUTINE USING SYSTBL.
;
SYSTBL:                         ; INDEX ADDRESS BY 0-7
        JMP     SYSTEM
        JMP     DO1
        JMP     DO2
        JMP     DO3
        JMP     DO4
        JMP     DO5
        JMP     DO6
        JMP     DO7
;
;THIS ROUTINE IS USED IN THE EVENT THE OPERATOR ENTERS AN INVALID
;CHARACTER AT THE CONSOLE.
;
ERROR:
        LXI     D,MSG1
        CALL    PRINT
        LXI     D,MENU
        CALL    PRINT
        JMP     AGAIN
;
;
;       START OF SYSTEM ROUTINES
;
;EACH ROUTINE CORRESPONDS TO THE NUMBER SELECTED IN THE MENU.  EACH MAIN
;ROUTINE IS NAMED IN THE FOLLOWING FORMAT DO'#', WHERE '#' IS THE NUMBER
;SELECTED ON THE MENU.  ALL SUPPORT ROUTINES TO THE MAIN ROUTINE WILL BE
;DESIGNATED BY DO'#'(ID), WHERE ID IS THE SPECIFIC SUPPORT ROUTINE
;DESIGNATOR.
;
;
DO1:
```

```
        LXI     D,MSG2
        CALL    PRINT
        CALL    CONIN
        MOV     C,A
        CALL    CONOUT
        MOV     A,C
        CPI     '0'
        JM      ERR1
        CPI     '4'
        JP      ERR1
        SUI     '0'
        MOV     C,A
        ADD     A
        ADD     C
        MOV     C,A
        XRA     A
        MOV     B,A
        LXI     E,DO1TBL
        DAD     B
        PCHL
;
;DO1TBL IS THE BASE ADDRESS FOR THE JUMP ROUTINE THAT SETS SPECIFIC
;SAMPLING RATES.  A SPECIFIC JUMP IS SELECTED BY THE DO1 ROUTINE
;USING DO1TBL AS THE BASE ADDRESS.
;
DO1TBL:
        JMP     DO10
        JMP     DO11
        JMP     DO12
        JMP     DO13
;
;DO10, DO11, DO12, DO13, ARE THE SPECIFIC ROUTINES SELECTED BY THE DO1TBL
;JUMPS.  ONLY ONE IS INVOKED AT ANY ONE RUN.  THE HL REG PR IS LOADED
;WITH THE NUMBER OF CLOCK CYCLES BETWEEN SAMPLES.  THIS PROGRAM IS SET
;TO RUN WITH A CLOCK RATE OF 4MHz.
;
DO10:
        LXI     H,8000
        JMP     DONE1
DO11:
        LXI     H,4000
        JMP     DONE1
DO12:
        LXI     H,1500
        JMP     DONE1
DO13:
        LXI     E,800
        JMP     DONE1
;
;DONE1 LOADS THE OUTPUT ADDRESS OF THE D TO A INTO DTCA AND OUTPUTS THE
;NUMBER OF CLOCK CYCLES BETWEEN SAMPLES INTO THE COMMAND REGISTER OF THE
;8155.  IT WILL START IMMEDIATEDLY AFTER LOADING.  AT TERMINAL COUNT IT
;WILL SEND THE NECESSARY INTERRUPT PULSE TO THE 7.5 INTERRUPT OF THE
;8085.
;
```

```
DONE1:
        DI
        MVI     A,80H
        STA     DTCA
        MOV     A,I
        OUT     TLSB
        MOV     A,B
        OFI     0C0H
        OUT     TMSBTM
        MVI     A,0C0H
        OUT     TPPICS
        LXI     D,MENU
        CALL    PRINT
        JMP     AGAIN
;
;PRINTS TO THE CONSOLE WHEN AN UNAUTHORIZED CHARACTER IS RECEIVED
;"BAD ENTRY, TRY AGAIN!"
;
ERR1:
        LXI     D,MSG1
        CALL    PRINT
        JMP     DO1
;
;THE SECOND MAIN ROUTINE INITIALIZES A BUBBLE.  THE SPECIFIC BUBBLE IS
;SELECTED BY THE USER AT THE CONSOLE.  IT RETURNS TO THE CONSOLE WITH AN
;"OPERATION COMPLETE" OR "INCOMPLETE."  IN INCOMPLETE IT  WILL ALSO GIVE
;THE RESULTS OF THE BUBBLE STATUS REGISTER.  THIS ROUTINE USED BASICALLY
;THE SAME ROUTINE TO INPUT DATA FROM THE CONSOLE AS DOES ROUTINE DO1.
;
DO2:
        DI
        LXI     D,MSG6
        CALL    PRINT
        CALL    CONIN
        MOV     C,A
        CALL    CONOUT
        MOV     A,C
        CPI     '0'
        JM      ERR2
        CPI     '6'
        JP      ERR2
        SUI     '0'
        MOV     C,A
        ADD     A
        ADD     C
        MOV     C,A
        XRA     A
        MOV     B,A
        LXI     H,DO2TBL
        DAD     B
        PCHI
;
;DO2TBL IS THE BASE ADDRESS FOR THE JUMP ROUTINES THAT INITIALIZE A SPECIFIC
;BUBBLE.  THE SPECIFIC JUMP IS SELECTED BY THE DO2 ROUTINE USING DO2TBL
;AS THE BASE ADDRESS.  THIS PROGRAM IS CAPABLE OF SELECTING 1 OF 5 SPECIFIC
```

```
;BUBBLES.  FOR THE SPACE SHUTTLE DATA RECORDER THIS WILL BE ENHANCED TO
;SELECT 1 OF 24 SPECIFIC BUBLES.
;
DO2TBL:
        JMP     DO20
        JMP     DO21
        JMP     DO22
        JMP     DO23
        JMP     DO24
        JMP     DO25
;
;THIS ROUTINE TURNS OFF THE POWER LIGHT (LED) TO A SPECIFIC BUBBLE
;
DO20:
        LXI     H,00H
        SHLD    BUFNUM
        MVI     A,00H
        OUT     PPIPA
        JMP     DONE21
;
;INITIALIZATION ROUTINE FOR BUBBLE #1
;
DO21:
        LXI     H,00H
        MVI     A,21H
        JMP     DONE2
;
;INITIALIZATION ROUTINE FOR BUBBLE #2
;
DO22:
        LXI     H,21H
        MVI     A,42H
        JMP     DONE2
;
;INITIALIZATION ROUTINE FOR BUBBLE #3
;
DO23:
        LXI     H,42H
        MVI     A,04H
        JMP     DONE2
;
;INITIALIZATION ROUTIN FOR BUBBLE #4
;
DO24:
        LXI     H,03H
        MVI     A,08H
        JMP     DONE2
;
;INITIALIZATION ROUTINE FOR BUBBLE #5
;
DO25:
        LXI     H,04H
        MVI     A,10H
;
;DONE2 INITIALIZES THE BUBBLE CARD INDICATED BY REGISTER PARI H-L.  IT
```

```
;ALSO STORES THIS B-L IN BUBNUM TO BE USED LATER AS AN OFFSET POINTER
;IN OTHER ROUTINES.
;
DONE2:
        SHLD    BUBNUM
        CUT     PPIPA
        CALL    BIOTMP
        LXI     B,TABLES
        CALL    INBUBL
        MOV     B,A
        XRI     40H
        JNZ     INTERR
        LXI     D,MSG4
        CALL    PRINT
        JMP     DONE21
;
;THIS ROUTINE PRINTS "OP-FAILED" IF INITIALIZATION IN ROUTINE DONE2 DOES
;NOT WORK.
;
INTERR:
        MOV     A,B
        CALL    STATUS
        LXI     D,MSG5
        CALL    PRINT
;
;RETURNS TO MENU UPON INITIALIZATION.
;
DONE21:
        LXI     D,MENU
        CALL    PRINT
        JMP     AGAIN
;
;PRINTS TO CONSOLE WHERE AN UNAUTHORIZED CHARACTER IS RECEIVED IN DO2.
;RETURNS TO MENU TO INITIALIZE A SPECIFIC BUBBLE.
;
ERR2:
        LXI     D,MSG1
        CALL    PRINT
        JMP     DO2
;
;THIS ROUTINE RECEIVES DATA FROM THE ANALOG TO DIGITAL CONVERTER AND PLACES
;IT TEMPORARILY IN THE RAM.  THE RAM IS DIVIDED INTO TWO PAGES.  FROM THE RAM
;IT IS THEN MOVED INTO THE BUBBLE MEMORY IN BLOCKS OF 32 PAGES (2K BYTES),
;WHERE A PAGE IS EQUAL TO 64 BYTES.  THE BYTES ARE PLACED IN THE RAM BY THE
;8085, A BYTE AT A TIME.  AFTER THE 32 PAGES HAVE BEEN READ INTO THE RAM THE
;FLAG "PAGFUL" IS SET TO INDICATE A BLOCK IS READY FOR TRANSFER TO THE BUBBLE
;MEMORY.  THE CURRENT BUBBLE IS INSTRUCTED TO START TRANSFERING THROUGH
;THE DMA.  WHILE THE BLOCK TRANSFER OF PAGES IS OCCURING, THE NEXT PAGE OF
;DATA FROM THE A TO D CONVERTERS IS BEING TRANSFERED INTO THE OTHER PAGE OF
;RAM.  BY THE TIME THE BLOCK TRANSFER IS COMPLETE, THE NEXT PAGE WILL BE
;NEARLY READY FOR TRANSFER.  ONCE IT IS FILLED, A FLAG WILL BE SET AND ANOTHER
;BLOCK TRANSFER WILL BEGIN AND THE PREVIOUS PAGE OF RAM WILL BE REWRITTEN,
;THUS BEGINNING THE CYCLE OVER AGAIN.  THIS WLL CONTINUE UNTIL ALL BUBBLE
;CARDS HAVE BEEN WRITTEN.
;
```

```
DO3:                                    ;RECORD INPUT DATA
        DI
        LXI     SP,STACK
        CALL    BSINT
        LXI     H,DOI3
        MVI     A,0C3H
        STA     INTVEC
        SHLD    INTVEC+1
        LXI     H,DO3CHK
        STA     DOCHK
        SHLD    DOCHK+1
;
;DMA INITIAL SET UP
;
        MVI     A,0FH
        OUT     0FH
        OUT     0CH
        MVI     A,00H
        OUT     04H
        MVI     A,0E0H
        OUT     04H
        MVI     A,00H
        OUT     05H
        MVI     A,2EH
        OUT     05H
        MVI     A,4AH
        OUT     0BH
        MVI     A,80H
        OUT     08H
;
        LXI     H,0
        SHLD    BUFNUM
        CALL    BIOTMP
        LXI     H,TXCNT
        SHLD    COUNT
        LXI     H,0E000H
        SHLD    RAMADD
        MVI     A,01
        STA     LETNUM
        OUT     PPIPA
        MVI     A,0FFH
        OUT     PPIPB
        STA     DMANUM
;
;SET UP RAM TABLE WITH PARAMETRIC REGISTER INITIAL VALUES.
;
        LXI     H,RAMTABL
        MVI     M,20H
        INX     H
        MVI     M,10H
        INX     H
        MVI     M,25H
        INX     H
        MVI     M,00H
        INX     H
```

```
        MVI     M,00H
;
        CALL    CONIN
;
        MVI     A,0
        STA     PAGFUL
        MVI     A,0AH
        SIM
        EI
;
;THIS ROUTINE POLLS THE "PAGE FULL" FLAG TO SEE IF A PAGE IS READY
;FOR TRANSFER.
;IF IT IS READY, IT SENDS A WRITE COMMAND TO THE BUBBLE TO START
;TRANSFERRING A PAGE FROM RAM TO THE CURRENT BUBBLE MEMORY.
;
PAGPOL3:
        LDA     PAGFUL
        CPI     0FFH
        JNZ     PAGPOL3
;
        LXI     B,RAMTABL
        CALL    WRBUBL
        MVI     A,09H
        SIM
        EI
        MVI     A,0BH
        OUT     0FH
;
BPOL3:
        JMP     BPOL3
;
;WHEN RECORDING, THIS ROUTINE IS THE 6.5 INTERRUPT ROUTINE.  THE 6.5 IS
;INITIALIZED BY THE INTERRUPT OF THE BUBBLE CARDS UPON AN OP-COMPLETE OR
;ERROR.  IF AN ERROR, IT WILL INDICATE BUBBLE STATUS REGISTER ON THE CONSOLE.
;
DO3CHK:
        MVI     A,2BH
        SIM
        EI
        CALL    RDSTMP
        MOV     B,A
        ANI     40H
        CPI     40H
        JZ      RESET3
;
;IF ERROR PRINT "OP FAIL" AND STATUS.
;
        LXI     D,CRLF
        CALL    PRINT
        MOV     A,B
        CALL    STATUS
        LXI     D,MSG5
        CALL    PRINT
        JMP     GOAGAIN
;
```

```
;CHECK TO SEE IF THE CURRENT BUBBLE IS FINISHED.  IF SO, MOVE TO NEXT CARD.
;IF LAST CARD, GO TO "GOAGAIN" TO START OVER.  THIS NORMALLY IS CALLED FROM
;DOCHK3.
;
RESET3:
        MVI     A,0FB
        OUT     0FB
        LHLD    COUNT
        DCX     H
        SHLD    COUNT
        MOV     A,H
        ORA     L
        JNZ     AGAIN3
        MVI     A,20B
        CALL    WRCTMP
        LDA     BUBCARD
        DCR     A
        STA     BUBCARD
        JZ      DONE3
        LHLD    BUBNUM
        INX     H
        SHLD    BUBNUM
        LDA     LEDNUM
        RLC
        STA     LEDNUM
        OUT     PPIPA
        LDA     DMANUM
        RLC
        STA     DMANUM
        OUT     PPIPB
        CALL    BIOTMP
        JMP     AGAIN3
;
;THIS ROUTINE PRINTS "OPERATION COMPLETE" TO CONSOLE AND RETURNS TO MAIN
;MENU.
;
DONE3:
        LXI     D,CRLF
        CALL    PRINT
        MOV     A,B
        CALL    STATUS
        LXI     D,MSG4
        CALL    PRINT
        JMP     GOAGAIN
;
;SET UP FOR NEXT BLOCK TRANSFER TO BUBBLE BY PLACING PARAMETRIC BUBBLE REG-
;ISTER VALUES INTO RAM.  THESE VALUES ARE READ BY THE WRCTMP ROUTINE AND
;PLACED INTO THE APPROPRIATE REGISTERS.
;
AGAIN3:
        MVI     A,2
        STA     PAGFCL
        LXI     H,RAMTABL
        MVI     M,20B
        INX     H
```

```
        MVI     M,10H
        INX     H
        MVI     M,25H
        INX     H
        MVI     A,0EH
        CALL    WRCTMP
        CALL    RDITMP
        MOV     M,A
        INX     H
        CALL    RDITMP
        MOV     M,A
;
;SET UP DMA FOR TRANSFER TO BUBBLE BY LOADING CORRECT RAM LOCATION.
;
        OUT     0CH
        LHLD    RAMADD
        MVI     A,0
        OUT     04H
        MOV     A,H
        CPI     0E6H
        JNC     AGN31
        MVI     A,2E0H
        JMP     AGN32
AGN31:
        MVI     A,2E8H
AGN32:
        OUT     04H
        MVI     A,2
        OUT     05H
        MVI     A,0EH
        OUT     05H
        JMP     PAGPOL3
;
;THIS ROUTINE IS VECTORED TO WHEN A 7.5 INTERRUPT IS ACKNOWLEDGED AND THE
;RECORDER IS IN THE RECORD MODE.  THIS ROUTINE TAKES A BYTE FROM THE ANALOG
;TO DIGITAL CONVERTERS AND PLACES IT IS THE CORRECT RAM LOCATION INDICATED
;BY THE ADDRESS STORED IN RAMADD.  THIS ROUTINE IS JUMPED TO THROUGH LOADING
;OF THE "INVECT" LOCATION WITH THE ADDRESS OF DO13.
;
DOI3:                                   ;BUBBLE WRITE INTERRUPT SERVICE
        PUSH    PSW
        PUSH    H
        LHLD    RAMADD
        LDA     ATOD
        STA     DTCA
        MOV     M,A
        MVI     A,0C0H
        SIM
        MVI     A,040H
        SIM
        INX     H
        MOV     A,H
        CPI     0F0H
        JZ      DOI31
        CPI     0E6H
```

```
        JZ      DOI32
        SHLD    RAMADD
        POP     H
        POP     PSW
        EI
        RET
;
;THIS ROUTINE SETS THE PAGE FULL FLAG IF PAGE 1 IS FULL AND RESETS THE RAM
;ADDRESS VALUE IN RAMADD.
;
DOI31:
        LXI     H,0E000H
        SHLD    RAMADD
        MVI     A,0FFH
        STA     PAGFUL
        POP     H
        POP     PSW
        EI
        RET
;
;THIS ROUTINE SETS THE "PAGE FULL" FLAG IF PAGE 0 IS FULL AND RESETS
;THE RAM ADDRESS VALUE IN RAMADD.
;
DOI32:
        SHLD    RAMADD
        MVI     A,0FFH
        STA     PAGFUL
        POP     H
        POP     PSW
        EI
        RET
;
;THIS ROUTINE TAKES THE DATA FROM THE BUBBLE MEMORY CARDS AND SENDS IT TO THE
;DIGITAL TO ANALOG CONVERTERS.  IT DOES THIS BY FIRST PLACING A BLOCK OF DATA
;INTO THE RAM.  A BLOCK OF DATA IS 32 SIXTY-FOUR BYTE PAGES.  THE BLOCK IS
;MOVED BY THE DMA FROM THE BUBBLE TO THE RAM.  ONCE IN THE RAM, THE DATA IS
;THEN MOVED BY THE 8085, A BYTE AT A TIME,TO THE DIGITAL TO ANALOG CONVERTERS.
;THE SAME TYPE OF PROCEDURE USED IN RECORDING THE DATA IS USED WHEN INTERLEAV-
;ING PAGE0 AND PAGE1 OF THE RAM.  SINCE THE DMA CAN TRANSFER FROM THE BUBBLE
;FASTER THAN THE BYTES ARE BEING PLACED OUT ONTO THE D TO A, THERE IS ALWAYS A
;FULL PAGE OF DATA READY WHEN THE CURRENT PAGE IS FINISHED BEING READ OUT.
;MOST OF THE CODING IS IN SUPPORT OF CHANGING PAGES OF RAM DURING PLAYBACK.
;
DO4:                                ;PLAYBACK DATA
        DI
        LXI     SP,STACK
        CALL    BSINT
        LXI     H,DOI4
        MVI     A,0C3H
        STA     INTVEC
        SHLD    INTVEC+1
        LXI     H,DO4CHK
        STA     DOCHK
        SHLD    DOCHK+1
;
```

```
;DMA INITIAL SET-UP
;
        MVI     A,0FH
        OUT     0FH
        OUT     0CH
        MVI     A,00H
        OUT     04H
        MVI     A,20E0H
        OUT     04H
        MVI     A,00H
        OUT     05H
        MVI     A,10H
        OUT     05H
        MVI     A,46H
        OUT     0BH
        MVI     A,60H
        OUT     08H
;
        LXI     H,0
        SHLD    BUBNUM
        CALL    BIOTMP
        LXI     H,TXCNT-1
        SHLD    COUNT
        LXI     H,0E000H
        SHLD    RAMADD
        MVI     A,01
        STA     LEDNUM
        OUT     PPIPA
        MVI     A,0FFH
        OUT     PPIPB
        STA     DMANUM
;
;SET UP RAM TABLE WITH PARAMETRIC REGISTER INITIAL VALUES.
;
        LXI     H,RAMTABL
        MVI     M,40H
        INX     H
        MVI     M,10H
        INX     H
        MVI     M,25H
        INX     H
        MVI     M,00H
        INX     H
        MVI     M,00H
;
        CALL    CONIN
;
        MVI     A,0DH
        SIM
        LXI     B,RAMTABL
        CALL    RDBUBL
        EI
        MVI     A,0BH
        OUT     0FH
DO41:
```

```
        JMP     DO41
;
;THIS ROUTINE POLLS THE "PAGE FULL" FIAG TO SEE IF A PAGE IS READY FOR
;TRANSFER.  IF IT IS READY, IT SENDS A READ COMMAND TO THE BUBBLE TO
;START TRANSFERING A PAGE FROM BUBBLE TO RAM.
;
PAGPOL4:
        LDA     PAGFUL
        CPI     0FFH
        JZ      PAGPOL4
;
        LXI     B,RAMTABL
        CALL    RDBUBL
        MVI     A,08H
        SIM
        EI
        MVI     A,0BH
        OUT     0FH
BPOL4:
        JMP     BPOL4
;
;WHEN READING, THIS ROUTINE IS THE 6.5 INTERRUPT SERVICE ROUTINE.  THE 6.5 IS
;INITIALIZED BY THE INTERRUPT OF THE BUBBLE CARDS UPON AN OP-COMPLETE OR
;ERROR.  IF AN ERROR, IT WILL INDICATE BUBBLE STATUS ON THE CONSOLE.
;
DO4CHK:
        MVI     A,0BH
        SIM
        EI
        CALL    RDSTMP
        MOV     B,A
        ANI     40H
        CPI     40H
        JZ      RESET4
;
;IF ERROR PRINT "OP-FAIL" AND STATUS.
;
        LXI     D,CRLF
        CALL    PRINT
        MOV     A,B
        CALL    STATUS
        LXI     D,MSG5
        CALL    PRINT
        JMP     GOAGAIN
;
;CHECK TO SEE IF THE CURRENT BUBBLE IS FINISHED.  IF SO MOVE TO NEXT CARD.
;IF LAST CARD GO TO GOAGAIN TO START OVER.  THIS NORMALLY IS CALLED FROM
;DO4CHK.
;
RESET4:
        MVI     A,0FFH
        STA     PAGFUL
        MVI     A,CFH
        OUT     0FH
        LHLD    COUNT
```

```
        DCX     H
        SHLD    COUNT
        MOV     A,H
        ORA     L
        JNZ     AGAIN4
        MVI     A,20H
        CALL    WRCTMP
        LDA     BUBCARD
        DCR     A
        STA     BUBCARD
        JZ      DONE4
        LHLD    BUBNUM
        INX     H
        SHLD    BUBNUM
        LDA     LEDNUM
        RLC
        STA     LEDNUM
        OUT     PPIPA
        LDA     DMANUM
        RLC
        STA     DMANUM
        OUT     PPIPB
        Call    BICTMP
        JMP     AGAIN4
;
;THIS ROUTINE PRINTS "OPERATION COMPLETE" TO THE CONSOLE AND RETURNS TO
;THE MAIN MENU.
;
DONE4:
        LXI     D,CRLF
        CALL    PRINT
        MOV     A,E
        CALL    STATUS
        LXI     D,MSG4
        CALL    PRINT
        JMP     GOAGAIN
;
;SET UP FOR NEXT BLOCK TRANSFER FROM BUBBLE TO RAM BY PLACING BUBBLE
;PARAMETRIC REG VALUES INTO RAM.  THESE VALUES ARE READ BY THE RDSTMP
;ROUTINE AND PLACED INTO THE APPROPRIATE REGS.
;
AGAIN4:
        LXI     H,RAMTABL
        MVI     M,20H
        INX     H
        MVI     M,10H
        INX     H
        MVI     M,25H
        INX     H
        MVI     A,2EH
        CALL    WRCTMP
        CALL    RDDTMP
        MOV     M,A
        INX     H
        CALL    RDDTMP
```

```
        MOV     M,A
;
;SET UP DMA FOR TRANSFER TO RAM BY LOADING CORRECT RAM LOCATIONS.
;
        OUT     0C8H
        LHLD    RAMADD
        MVI     A,0
        OUT     04H
        MOV     A,H
        CPI     0E8H
        JNC     AGN41
        MVI     A,0E0H
        JMP     AGN42
AGN41:
        MVI     A,0E8H
AGN42:
        OUT     04H
        MVI     A,0
        OUT     05H
        MVI     A,08H
        OUT     05H
        JMP     PAGPOL4
;
;THIS ROUTINE IS VECTORED TO WHEN A 7.5 INTERRUPT IS ACKNOWLEDGED AND THE
;RECORDER IS IN THE PLAYBACK MODE.  THIS ROUTINE TAKES A BYTE FROM RAM AND
;PLACES IT ON THE CORRECT DIGITAL TO ANALOG CONVERTER.  THIS ROUTINE IS
;JUMPED TO THROUGH LOADING OF THE INVECT LOCATION WTH THE ADDRESS OF DOI4.
;
DOI4:                                   ;READ BUBBLE INTERRUPT SERVICE
        PUSH    PSW
        PUSH    H
        LELD    RAMADD
        MOV     A,M
        STA     DTOA
        INX     H
        MOV     A,H
        CPI     0F0H
        JZ      DOI41
        CPI     0E8H
        JZ      DOI42
        SHLD    RAMADD
        POP     H
        POP     PSW
        EI
        RET
;
;THIS ROUTINE SETS THE "PAGE FULL" FLAG IF THE CURRENT PAGE IS FULL AND
;RESETS THE RAM ADDRESS VALUE IN RAMADD.
;
DOI41:
        LXI     H,0E000H
        SHLD    RAMADD
        MVI     A,0BH
        STA     PAGFUL
        POP     H
```

```
        POP     PSW
        EI
        RET
;
;THIS ROUTINE ALSO SETS THE "PAGE FULL" FLAG FOR THE OTHER PAGE IN RAM AND
;RESETS THE RAM ADDRESS VALUE IN RAMADD.
;
DOI42:
        SHLD    RAMADD
        MVI     A,00H
        STA     PAGFUL
        POP     H
        POP     PSW
        EI
        RET
;
;THIS ROUTINE IS REFERENCED WHEN AN ERROR OCCURS.  "OP-FAIL"IS PRINTED OUT
;ONTO THE CONSOLE ALONG WITH THE STATUS OF THE BUBBLE MEMORY CONTROLLER
;SO THAT THE USER IS ABLE TO DETERMINE WHAT WENT WRONG WITH THE BUBBLE
;READ OR WRITE OPERATION.
;
STATUS:                         ;PRINT BUBBLE CONTROLLER STATUS
        MVI     B,08H
STAT1:
        RLC
        JC      STAT2
        MVI     C,30H
        JMP     STAT3
STAT2:
        MVI     C,31H
STAT3:
        PUSH    PSW
        CALL    CONOUT
        POP     PSW
        DCR     B
        JNZ     STAT1
        RET
;
;THIS NEXT ROUTINE ENABLES THE PROGRAM TO RETURN TO THE MAIN MENU
;FROM A READ OR WRITE OPERATION.
;
GOAGAIN:                        ;RETURN TO SYSTEM FROM READ/WRITE
        DI
        MVI     A,0FH
        OUT     0FH
        MVI     A,60H
        OUT     06H
        MVI     A,60H
        STA     DTOA
        MVI     A,0
        OUT     PPIPA
        MVI     A,0FFH
        OUT     PPIPB
        LXI     D,MENU
        CALL    PRINT
```

```
        JMP     AGAIN
;
;BSINT AND ITS SUBSEQUENT ROUTINES, BSINT0 AND BSINT1 RESPECTIVELY, ARE
;DEALING WITH THE INITIALIZATION OF THE 6 BUBBLE CARDS.  WHENEVER AN
;INITIALIZATION IS PERFORMED, THESE ROUTINES FURNISH THE USER (VIA THE
;CONSOLE) SUCH INFORMATION AS TO HOW MANY CARDS INITIALIZED SUCCESSFULLY.
;
BSINT:
        LXI     H,00H
        MVI     A,01H
        CALL    BSINT1
        LXI     H,01H
        MVI     A,02H
        CALL    BSINT1
        LXI     H,02H
        MVI     A,04H
        CALL    BSINT1
        LXI     H,03H
        MVI     A,08H
        CALL    BSINT1
        LXI     H,04H
        MVI     A,10H
        CALL    BSINT1
BSINT0:
        STA     BUBCARD
        LXI     D,CRLF
        CALL    PRINT
        LDA     BUBCARD
        ADI     30H
        MOV     C,A
        CALL    CONOUT
        LXI     D,MSG10
        CALL    PRINT
        LXI     H,00H
        SHLD    BUBNUM
        MVI     A,00H
        OUT     PPIPA
        RET
BSINT1:
        SHLD    BUBNUM
        OUT     PPIPA
        CALL    BIOTMP
        LXI     B,TABLES
        CALL    INBUBL
        MOV     B,A
        MVI     A,20H
        CALL    WRCTMP
        MOV     A,B
        ANI     0F4H
        CPI     40H
        MVI     A,05H
        RZ
        LHLD    BUBNUM
        MOV     A,L
        POP     B
```

```
        JMP     B5INT0
;
;DOTRAP IS USED WHEN A SPURIOUS INTERRUPT OCCURS WITHIN THE SYSTEM
;THAT COULD POSSIBLY RUIN THE CURRENT OPERATION.  THE WAY IN WHICH THE
;USER LEARNS THAT SOMETHING HAPPENED, I.E., AN UNEXPECTED INTERRUPT,
;IS THAT THE SYSTEM JUMPS TO THIS ROUTINE AND PRINTS OUT "SPURIOUS
;INTERRPUT" ONTO THE CONSOLE.
;
DOTRAP:
        LXI     D,MSG7
        CALL    PRINT
        JMP     AGAIN
;
;THE NEXT SET OF ROUTINES, DO5 THROUGH DO155, ARE USED FOR THE DATA ERROR
;TESTING TO ENSURE PROPER OPERATION OF THE BUBBLES.  THE PREDICTED ERROR
;RATE FOR THE BUBBLES IS 10^-13.  IF ONE DESIRED TO, THEY COULD RUN THIS
;ROUTINE INDEFINTELY TO SEE IF THAT PREDICTION HOLDS.  DURING THIS SECTION
;DIFFERENT PATTERNS OF "1'S" AND "0'S" ARE RECORDED INTO THE BUBBLE MEM-
;ORIES.  THE USER HAS A CHOICE OF WHICH PATTERN (SEE MSG11).  UPON COM-
;PLETION, THE USER CAN PLAYBACK WHAT WAS RECORDED AND DETERMINE THE
;ACCURACY OF THE BUBBLE MEMORIES.
;
DO5:
        DI
        LXI     SP,STACK
        LXI     D,MSG11
        CALL    PRINT
        CALL    B5INT
DO500:                          ;CONSOLE ENTRY SECTION.
        CALL    CONIN
        MOV     C,A
        CALL    CONOUT
        MOV     A,C
        CPI     '0'
        JM      ERR5
        CPI     '5'
        JP      ERR5
        SUI     '0'
        MOV     C,A
        ADD     A
        ADD     C
        MOV     C,A
        XRA     A
        MOV     B,A
        LXI     H,DO5TBL
        DAD     B
        PCHL
DO5TBL:
        JMP     DO50
        JMP     DO51
        JMP     DO52
        JMP     DO53
        JMP     DO54
ERR5:                           ;ERROR ROUTINE FOR INCORRECT  CONSOLE ENTRY
        LXI     D,MSG1
```

```
        CALL    PRINT
        LXI     D,MSG11
        CALL    PRINT
        JMP     D0500
D050:
        JMP     GOAGAIN
D051:
        JMP     DONE5
D052:
        JMP     DONE5
D053:
        JMP     DONE5
D054:
        JMP     DONE5
DONE5:                          ;ROUTINE TO PRINT CHOICES FOR DATA ERROR TEST.
        LXI     D,MSG11
        CALL    PRINT
        JMP     D0500
;
D0155:                          ;MSG TO CONSOLE INCASE OF SPURIOUS INTERRUPT.
        LXI     D,MSG7
        CALL    PRINT
        JMP     AGAIN
;
;THIS SECTION, D06-D061, IS DESIGNED TO CLEAR THE RAM BUFFER IF CHOSEN.
;IT IS WISE TO START OUT WITH A RAM THAT IS IN A KNOWN STATE.  THEN THE
;USER IS ASSUMED TO HAVE NO PREVIOUS DATA FROM OTHER RECORDINGS.
;UPON COMPLETION IT RETURNS THE USER TO THE MAIN MENU.
;
D06:
        DI
        LXI     H,PAGE0
        LXI     B,1000
        MVI     D,80H
D061:
        MOV     M,D
        DCX     B
        MOV     A,B
        ORA     C
        JNZ     D061
        MVI     A,80H
        STA     DTCA
        LXI     D,MSG4
        CALL    PRINT
        LXI     D,MENU
        CALL    PRINT
        JMP     AGAIN
;
;DC7 AND D017 ARE THE CONVERT AND DISPLAY OPTIONS.  THE ADVANTAGE OF THESE
;ROUTINES IS THAT THE USER CAN ENSURE THE CORRECT OPERATION OF THE ANALOG
;AND DIGITAL CIRCUITRY.  NO RECORDING OR PLAYING BACK IS PERFORMED.  THE
;SAMPLED INPUT GOES DIRECTLY TO THE OUTPUT.  HENCE, ONE IS ABLE TO CONNECT
;THE INPUT AND OUTPUT TO AN OSCILLOSCOPE, FOR EXAMPLE, AND DETERMINE
;THE ACCURACY OF BOTH WAVEFORMS.  THE ONLY DISCREPANCY THAT MAY BE
;OBSERVED WILL BE A TIME DELAY.
```

```
;
DO7:                                    ;LOADING OF THE REGISTERS.
        DI
        LXI     H,DOI7
        MVI     A,0C3H
        STA     INTVEC
        SHLD    INTVEC+1
        MVI     A,2FH
        OUT     0FH
        OUT     0CH
        MVI     A,00H
        OUT     00H
        MVI     A,0C0H
        OUT     00H
        MVI     A,21H
        OUT     01H
        MVI     A,00H
        OUT     01H
        MVI     A,01H
        OUT     02H
        MVI     A,2C0H
        OUT     02H
        MVI     A,01H
        OUT     03H
        MVI     A,00H
        OUT     03H
        MVI     A,98H
        OUT     0BH
        MVI     A,95H
        OUT     0BH
        MVI     A,61H
        OUT     02H
        MVI     A,2EH
        OUT     0FH
        MVI     A,0CBH
        SIM
        MVI     A,040H
        SIM
        EI
        LXI     D,MENU
        CALL    PRINT
        JMP     AGAIN
;
DOI7:                                   ;CONVERT AND DISPLAY INTERRUPT ROUTINE
        PUSH    PSW
        PUSH    H
        PUSH    D
        PUSH    B
        MVI     A,04H
        OUT     09H
        NOP
        MVI     A,0C0H
        SIM
        NOP
        MVI     A,040H
```

```
        SIM
        POP     B
        POP     D
        POP     H
        POP     PSW
        EI
        RET
;
;
;
;       START OF SUBROUTINES
;
;THE FOLLOWING SUBROUTINES SET THE HARDWARE TO KNOWN CONDITIONS AND
;PROVIDE THE CODE FOR PRINTING OUT MENUS AND MESSAGES VIA THE CONSOLE.
;EVERYTIME A MESSAGE IS TO BE PRINTED, THE PRINT ROUTINES ARE CALLED.
;EVERYTIME THE USER IS REQUIRED TO INPUT (OR RECEIVED AN OUTPUT) VIA
;THE CONSOLE, THE CONIN/CONOUT/CONST SUBROUTINES ARE CALLED.
;INITHW IS THE SUBROUTINE THAT IS CALLED TO SET ALL THE HARDWARE TO A
;KNOWN STATE.
;
PRINT:
        XCHG
PRT1:
        MOV     A,M
        CPI     '$'
        RZ
        MOV     C,A
PRT2:
        IN      CONSTAT
        ANI     01
        CPI     01
        JNZ     PRT2
        MOV     A,C
        OUT     CONDATA
        INX     H
        JMP     PRT1
;
CONIN:
        IN      CONSTAT
        ANI     02
        JZ      CONIN
        IN      CONDATA
        ANI     7FH
        RET
;
CONOUT:
        IN      CONSTAT
        ANI     01
        CPI     01
        JNZ     CONOUT
        MOV     A,C
        OUT     CONDATA
        RET
;
CONST:
```

```
        IN      CONSTAT
        ANI     02
        RZ
        MVI     A,0FFH
        RET
;
INITHW:                         ;SET HARDWARE TO KNOWN STATE
;
;8251 UART
;
        MVI     A,0
        OUT     CONSTAT
        OUT     CONSTAT
        OUT     CONSTAT
        MVI     A,40H
        OUT     CONSTAT
        MVI     A,4EH
        OUT     CONSTAT
        MVI     A,37H
        OUT     CONSTAT
;
;8255 PROGRAMMABLE PERIPHERAL INTERFACE.
;
        MVI     A,80H
        OUT     PPICS
        MVI     A,00H
        OUT     PPIPA
        MVI     A,0FFH
        OUT     PPIPB
;
;ZERO A TO D STROBE
;
        MVI     A,48H
        SIM
;
;ZERO D TO A OUTPUT
;
        MVI     A,80H
        STA     DTOA
;
;8237 DMA
;
        MVI     A,0
        OUT     0DH
;
;RESET ALL REGISTERS
;
        OUT     00
        OUT     00
        OUT     01
        OUT     01
        OUT     02
        OUT     02
        OUT     03
        OUT     03
```

```
        CUT     04
        CUT     04
        OUT     05
        OUT     05
        CUT     06
        OUT     06
        CUT     07
        OUT     07
;
;SET MODE REGISTERS
;
        MVI     A,9EH
        OUT     0BH
        MVI     A,95H
        CUT     0BH
        MVI     A,16H
        OUT     0BH
        MVI     A,1BH
        OUT     0BH
;
;SET COMMAND REGISTER
;
        MVI     A,61H
        OUT     0EH
        RET
;
DELAY:
        MOV     B,A             ;DELAY A TIMES 10MSEC.
LOOP1:
        LXI     D,1666
LOOP2:
        DCX     D
        MOV     A,D
        ORA     E
        JNZ     LOOP2
        DCR     B
        JNZ     LOOP1
        RET
;
;THIS IS THE MAIN MENU AND CHOICE OF OPERATIONS THAT CAN BE PERFORMED.
;
MENU:   DB      CR,LF,'SOLID STATE DATA RECORDER',CR,LF
        DB      CR,LF,'0= RESET SYSTEM '
        DB      CR,LF,'1= SET SAMPLE RATE '
        DB      CR,LF,'2= INITIALIZE BUBBLE '
        DB      CR,LF,'3= RECORD INPUT DATA '
        DB      CR,LF,'4= PLAYBACK DATA '
        DB      CR,LF,'5= DATA ERROR TEST '
        DB      CR,LF,'6= CLEAR RAM '
        DB      CR,LF,'7= CONVERT / DISPLAY '
        DB      CR,LF,'$'
;
;THE FOLLOWING MESSAGES ARE SELF-EXPLANATORY AND ARE CALLED THROUGHOUT THE
;PROGRAM IN ORDER FOR THE SYSTEM TO COMMUNICATE WITH THE OPERATOR VIA THE
;CONSOLE.
```

```
;
MSG1:   DB      CR,LF,'BAD ENTRY, TRY AGAIN!',CR,LF,'$'
;
MSG2:   DB      CR,LF,'SET SAMPLE RATE ',CR,LF
        DB      CR,LF,'0= 500HZ '
        DB      CR,LF,'1= 1.0KHZ '
        DB      CR,LF,'2= 2.5KHZ '
        DB      CR,LF,'3= 5.0KHZ '
        DB      CR,LF,'$'
;
MSG3:   DB      CR,LF,'SPARE ',CR,LF,'$'
;
MSG4:   DB      CR,LF,'OP-COMPLETE ',CR,LF,'$'
;
MSG5:   DB      CR,LF,'OP-FAILED ',CR,LF,'$'
;
MSG6:   DB      CR,LF,'CHOOSE BUBBLE FOR INITIALIZATION',CR,LF
        DB      CR,LF,'0= CLEAR'
        DB      CR,LF,'1= BUBBLE 1'
        DB      CR,LF,'2= BUBBLE 2'
        DB      CR,LF,'3= BUBBLE 3'
        DB      CR,LF,'4= BUBBLE 4'
        DB      CR,LF,'5= BUBBLE 5'
        DB      CR,LF,'$'
;
MSG7:   DB      CR,LF,'SPURIOUS INTERRUPT ',CR,LF,'$'
;
MSG8:   DB      CR,LF,'TIME OUT ERROR ',CR,LF,'$'
;
MSG9:   DB      CR,LF,'SPARE',CR,LF,'$'
;
MSG10:  DB      ' CARDS ACTIVE',CR,LF,'$'
;
MSG11:  DB      CR,LF,'SELECT TEST MODE',CR,LF
        DB      CR,LF,'0= END TEST '
        DB      CR,LF,'1= 00000000 '
        DB      CR,LF,'2= 01010101 '
        DB      CR,LF,'3= 10101010 '
        DB      CR,LF,'4= 11111111 '
        DB      CR,LF,'$'
;
CRLF:   DB      CR,LF,'$'
;
;THESE ARE THE INITIAL VALUES FOR THE PARAMETRIC REGISTERS.  THEY ARE
;LOADED INTO RAM INITIALLY SO THAT THERE IS A CAPABILITY TO CHANGE THE
;VALUES THROUGHTOUT THE OPERATION OF THE RECORDER.
;
TABLES: DB      01B,10H,20H,00H,00H
;
        DS      1
;
        END
```

```
;
;THIS SECTION WAS LINKED ONTO THE MAIN PROGRAM AND STORED IN THE EPROM.
;IT IS THEN COPIED FROM EPROM INTO THE RAM IN ORDER TO HAVE THE CAPABILITY
;TO CHANGE THE PARAMETRIC REGISTER VALUES DURING A RECORD OR PLAYBACK OPERA-
;TION.  OBVIOUSLY, TO WORK OUT OF THE EPROM, THE VALUES WOULD NEVER HAVE
;BEEN ABLE TO BE CHANGED.  CONSEQUENTLY, "TABLES" CONTAINS THE INTITIAL
;VALUES BUT THE RAM COPY  IS THE PART THAT CAN BE UPDATED TO REFLECT CURRENT
;VALUES.
;
ENTRY   TABLES,BUB0,BUB1,BUB2,BUB3,BUB4,ENDTAB
;
        CSEG
;
;       SOLID STATE DATA RECORDER PROGRAM
;       FOR 4-MEGABIT BUBBLE MEMORY BOARD
;
STACK   EQU     0
;
PRTA00  EQU     0FD00H
PRTA01  EQU     0FD01H
;
TPPICS  EQU     10H
TPPIPA  EQU     11H
TPPIPB  EQU     12H
TPPIPC  EQU     13H
TLSB    EQU     14H
TMSBTM  EQU     15H
;
PPIPA   EQU     20H
PPIPB   EQU     21H
PPIPC   EQU     22H
PPICS   EQU     23H
;
CONDATA EQU     30H
CONSTAT EQU     31H
;
ATCD    EQU     0C000H
DTCA    EQU     0C001H
;
CR      EQU     0DH
LF      EQU     0AH
BS      EQU     08H
;
PAGE0   EQU     0E000H
PAGE1   EQU     0E100H
;
RAM     EQU     0F000H
RAMTABL EQU     RAM
ATCDTMP EQU     RAMTABL+100H
DTOATMP EQU     ATODTMP+2
;
;       MOVE TABLE
;
TABLES: DB      01H,10H,20H,00H,00H
BUB0:   DB      01H,10H,20H,00H,20H
```

```
BUF1:   DB      01H,10H,20H,00H,00H
BUF2:   DB      01H,10H,20H,00H,00H
BUF3:   DB      01H,10H,20H,00H,00H
BUF4:   DB      01H,10H,20H,00H,00H
ENDTAB  EQU     $
TABSIZ  EQU     ENDTAB-TABLES
;
        DS      1
;
        END
```

```
;
;       4-MEG BUBBLE DRIVERS
;
;THE SOFTWARE DRIVERS FOR THE BUBBLE MEMORIES ARE SUPPLIED BY INTEL
;CORPORATION.  ULMONT SMITH WAS THE ORIGINAL AUTHOR OF THE DRIVERS FOR THE
;ONE MEGABIT DEVICE.  MOST OF THE SAME ONES CAN BE USED ON THE FOUR MEGA-
;BIT, HOWEVER, CHANGES NEEDED TO BE INCORPORATED IN ORDER FOR THEM TO
;RUN THE FOUR MEGABIT MEMORIES WITHOUT ERRORS.  ONE OF THE ADDITIONS WAS
;THE ADDITION OF THE DRIVER TO WRITE ALL ONE'S TO THE FIFO/BOOTLOOP IN
;ORDER TO FORCE THE CONTROLLER TO UTILIZE THE  EXTERNAL DATA ERROR CHECKING.
;THE OTHER ADDITION WAS THAT OF THE BIOTMP ROUTINE SO THAT THE DRIVERS DID
;NOT HAVE TO BE REPEATED 6 TIMES (IN THE CASE OF THE SSDR, 24 TIMES.)
;
;
;
EXTRN   TABLES,BUBNUM,RDDTMP,WRDTMP,RDSTMP,WRCTMP
;
ENTRY   INBUBL,RDBUBL,WRBUBL
;
BYTCNT  EQU     256
;
;INTPAR INITIALIZES THE PARAMETRIC REGISTERS OF THE BUBBLES.  THESE REGISTERS
;MUST BE LOADED EACH TIME A READ OR A WRITE OPERATION IS PERFORMED.
;
; NOTE:  THIS DOES DESTROY THE A, F/FS
;
INTPAR: PUSH    B       ;SAVE B-C REGISTERS
        PUSH    D       ;SAVE D-E REGISTERS
        MVI     A,2BH   ;LOAD A REG WITH BLR LSB ADDRESS
        CALL    WRCTMP
        MVI     E,05H   ;INITIALIZE LOOP COUNTER
LOAD:   LDAX    B       ;LOAD A REG FROM B-C REG ADDRESS
        CALL    WRDTMP
        INX     B       ;INCREMENT B-C REGS TO THE NEXT ADDR IN RAM
        DCR     E       ;DECREMENT LOOP COUNTER
        JNZ     LOAD    ;IF NOT ZERO, JMP LOAD
        POP     D       ;RESTORE D-E REGS
        POP     B       ;RESTORE B-C REGS
        RET
;
;
;       RESET 7220 FIFO DATA BUFFER
;
; NOTE: DESTROYS A, F/FS
;
FIFORS: PUSH    D       ;SAVE D-E REGS
        PUSH    B       ;SAVE B-C REGS
        MVI     B,40H   ;LOAD B REG OP-COMPLETE
        LXI     D,0FFFFH;INTIALIZE TIME OUT
        MVI     A,1DH   ;LOAD A REG RESET COMMAND
        CALL    WRCTMP
BUSYFR:
        CALL    RDSTMP
        RLC             ;TEST BUSY BIT=1
```

```
        JC      POLLFR  ;IF BUSY=1,POLL STATUS REG
        DCX     D       ;DECREMENT TIME OUT LOOP
        XRA     A       ;CLEAR A REG
        ORA     D       ;TEST D-REG=00H
        ORA     E       ;TEST E-REG=00H
        JNZ     BUSYFR  ;IF NOT 0 CONTINUE POLLING
        JMP     RETFR   ;TIME OUT ERROR
POLLFR:
        CALL    RDSTMP
        XRA     B       ;TEST STATUS = 40H
        JZ      RETFR   ;IF OP-COMPLETE JMP RETFR
        DCX     D       ;DECREMENT TIME  OUT LOOP
        XRA     A       ;CLEAR A
        ORA     D       ;TEST D-REG
        ORA     E       ;TEST E-REG
        JNZ     POLLFR  ;IF NOT 0 CONTINUE POLLING
RETFR:  POP     B       ;RESTORE B-C REG
        POP     D       ;RESTORE D-E REG
        JMP     RDSTMP
;
;
;ABORT MUST BE PERFORMED WHEN A BUBBLE IS INITIALLY TURNED ON OR IMMEDIATELY
;BEFORE POWERING DOWN THE BUBBLE.
;
;       NOTE: DESTROYS A, F/FS
;
ABORT:  PUSH    D       ;SAVE D-E REGS
        PUSH    B       ;SAVE B-C REGS
        LXI     D,0FFFFH;INIT TIME OUT LOOP COUNTER
        MVI     B,40H   ;LOAD B REG= 40H, OP-COMPLETE
        MVI     A,19H   ;LOAD A REG= ABORT COMMAND
        CALL    WRCTMP
BUSYA:
        CALL    RDSTMP
        RLC             ;TEST BUSY BIT= 1
        JC      POLLA   ;IF BUSY= 1, POLL STATUS REG FOR 40H
        DCX     D       ;DEC TIME OUT LOOP COUNTER
        XRA     A       ;CLEAR A REG
        ORA     D       ;TEST D REG= 00H
        ORA     E       ;TEST E REG= 00H
        JNZ     BUSYA   ;IF NOT 0, CONT POLLING ABORT COMMAND
        JMP     RETA    ;TIME OUT ERROR, RETURN
POLLA:
        CALL    RDSTMP
        XRA     B       ;TEST STATUS =40H, OP-COMPLETE
        JZ      RETA    ;IF OP-COMPLETE, JMP RETA
        DCX     D       ;DEC TIME OUT LOOP COUNTER
        XRA     A       ;TEST STATUS FOR OP-COMPLETE
        ORA     D       ;TEST D REG FOR 0
        ORA     E       ;TEST E REG FOR 0
        JNZ     POLLA   ;IF NOT 0 CONTINUE POLLING
RETA:   POP     B       ;RESTORE B-C REGS
        POP     D       ;RESTORE D-E REGS
        JMP     RDSTMP
;
```

```
;
;WRITE 7224 FIFO DATA BUFFER WITH ALL ONE'S.  THIS PARTICULAR DRIVER HAD TO
;BE WRITTEN BECAUSE THE RECORDER IS USING THE 4MBIT DEVICES.  WITH THE FIRST
;IN THE SERIES OF 4MBIT BUBBLES, THE BOOTLOOP HAD TO BE REWRITTEN (OR APPEAR
;TO BE) BY WRITING ALL ONE'S TO THE FIFO.  THEN WHEN THE CONTROLLER DOES A
;COMPARE AND "SEES" ALL ONE'S--THEREFORE, IT INTERPRETS THAT AS A BUBBLE
;WITH ALL LOOPS ACTIVE--IT GOES OUT AND DOES AN EXTERNAL DATA ERROR CHECK.
;
; DESTROYS A, F/FS
;
WRFIFO:                                 ;FILL FIFO WITH ALL ONE'S
        PUSH    B
        PUSH    D
        MVI     B,40H
        MVI     C,28H
        CALL    FIFOBS
        XRA     B
        JNZ     RETWF
        MVI     A,0FFH
INFIFO:
        CALL    WRDTMP
        DCR     C
        JNZ     INFIFO
RETWF:
        POP     D
        POP     B
        JMP     RDSTMP
;
;
;WRITE 7244 BOOT LOOP REGISTERS WITH ALL ONE'S.  THIS IS THE SECOND HALF OF
;THE PROCESS OF FAKING THE BUBBLE INTO BELIEVING THE BOOTLOOP HAS ALL ONE'S
;AND THEREFORE MUST DO AN EXTERNAL DATA ERROR CHECK.
;
; DESTROYS A, F/FS
;
WRBLRS:
        PUSH    B
        PUSH    H
        MVI     B,41H
        MVI     C,0FDH
        LXI     H,0
        CALL    WRFIFO
        ANA     C
        XRA     B
        JNZ     RETWBL
        DCR     B
        MVI     A,16H
        CALL    WRDTMP
BSYWBL:
        CALL    RDSTMP
        RLC
        JC      BSYWBL
        DCX     H
        MOV     A,H
        ORA     L
```

```
        JNZ     BSYWBL
        JMP     RETWBI
POLWBL:
        CALL    RDSTMP
        XRA     B
        JZ      RETWBL
        DCX     H
        MOV     A,H
        ORA     L
        JNZ     POLWBL
RETWBL:
        POP     H
        POP     B
        JMP     RDSTMP
;
;
;WRITE BUBBLE MEMORY DATA IS THE ROUTINE USED TO PREPARE THE BUBBLE FOR A
;WRITE OPERATION.
;
; DESTROYS A, F/FS
;
WRBUBL: ;WRITE BUBBLE DATA
        RET
        PUSH    H       ;SAVE H-L REG
        PUSH    B       ;SAVE B-C REG
        PUSH    D
        CALL    BIOTMP
        MVI     B,40H   ;LOAD B REG OP-COMPLETE
        CALL    FIFORS  ;RESET FIFO
        XRA     B       ;TEST FOR OP-COMPLETE
        JNZ     RETWR   ;IF ERROR JMP RETWR
        POP     B       ;RESTORE B-C
        CALL    INTPAR  ;LOAD PARAMETRIC REGS
        LXI     H,BYTCNT
        PUSH    B       ;SAVE B-C REGS
        LXI     H,0FFFFH;INITIALIZE TIME OUT LOOP
LOOPWR:
        CALL    RDSTMP
        RLC             ;TEST   FOR BUSY=1
        JNC     RETWR   ;IF ZERO JMP RETWR
        DCX     H       ;DECREMENT TIME OUT LOOP
        XRA     A       ;CLEAR A REG
        ORA     H       ;TEST H-REG FOR 0
        ORA     L       ;TEST L-REG FOR 0
        JNZ     LOOPWR  ;CONTINUE POLLING
RETWR:
        POP     D
        POP     B
        POP     H       ;RESTORE H-L REGS
        JMP     RISTMP
;
;
;READ BUBBLE MEMORY DATA IS THE ROUTINE USED TO PREPARE THE BUBBLE FOR
;A READ OPERATION.
;
```

```
; DESTROYS A, F/FS
;
RDBUBL: ;READ BUBBLE DATA
        RET
        PUSH    H       ;SAVE H-I REGS
        PUSH    B       ;SAVE B-C REGS
        PUSH    D
        CALL    BIOTMP
        MVI     B,40H   ;LOAD B REG OP-COMPLETE
        CALL    FIFORS  ;RESET FIFO
        XRA     B       ;TEST FOR OP-COMPLETE
        JNZ     RETRD   ;IF NOT ZERO JMP RETRD
        POP     B       ;RESTORE B-C REGS
        CALL    INTPAR  ;LOAD PARAMETRIC REGS
        LXI     H,BYTCNT
        PUSH    B       ;SAVE B-C REGS
        LXI     H,0FFFFH;INITIALIZE TIME OUT LOOP
LOOPRD:
        CALL    RDSTMP
        RLC             ;TEST FOR BUSY=1
        JNC     RETRD   ;IF ZERO,NOT BUSY,JMP RETRD
        DCX     H       ;DECREMENT TIME OUT LOOP
        XRA     A       ;CLEAR A REG
        ORA     H       ;TEST H REG=0
        ORA     L       ;TEST L REG=0
        JNZ     LOOPRD  ;CONTINUE POLLING
RETRD:
        POP     D
        POP     B
        POP     H       ;RESTORE H-I REGS
        JMP     RDSTMP
;
;
;INITIALIZE THE BUBBLE MUST BE PERFORMED EVERYTIME THE BUBBLE IS TO BE
;WRITTEN TO OR READ FROM.  IT SETS UP THE BUBBLE INTO A KNOWN STATE.
;
; DESTROYS A, F/FS
;
INBUBL:
        PUSH    D
        PUSH    B
        CALL    BIOTMP
        MVI     B,40H   ;LOAD B REG OP-COMPLETE
        CALL    ABORT   ;CALL ABORT COMMAND
        XRA     B       ;TEST FOR OP-COMPLETE
        JNZ     RETIN   ;IF ZERO OP-COMPLETE
        POP     B       ;ADDRESS OF PARAMETRIC REGS
        CALL    INTPAR  ;LOAD PARAMETRIC REGS
        PUSH    B       ;SAVE B-C REGS
        MVI     B,40H   ;LOAD B REG OP-COMPLETE
        LXI     D,0FFFFH;INITIALIZE TIME OUT LOOP
        MVI     A,11H   ;LOAD A REG INIT COMMAND
        CALL    WRCTMP
BUSYIN:
        CALL    RDSTMP
```

```
        RLC             ;DECREMENT TIME OUT LOOP
        JC      POLLIN  ;IF BUSY=1 POLL FOR 40E
        DCX     D       ;DECREMENT TIME OUT LOOP
        XRA     A       ;CLEAR A REG
        ORA     D       ;TEST D REG FOR 0
        CRA     E       ;TEST E REG FOR 0
        JNZ     BUSYIN  ;IF NOT 0 CONTINUE POLLING
        JMP     RETIN   ;TIME OUT ERROR, RETURN
POLLIN:
        CALL    RDSTMP
        XRA     B       ;TEST FOR OP-COMPLETE
        JZ      EARFIX
        DCX     D       ;DECREMENT TIME OUT LOOP
        XRA     A       ;CLEAR A REG
        CRA     D       ;TEST D REG FOR 0
        ORA     E       ;TEST E REG FOR 0
        JNZ     POLLIN  ;IF NOT 0 CONTINUE POLLING
EARFIX:
        CALL    WRBLRS
RETIN:
        POP     B
        POP     D
        JMP     RDSTMP
;
;BIOTMP AND ITS SUBSEQUENT ROUTINES SET UP THE POINTERS SO THAT THE BUBBLE
;DRIVERS DON'T HAVE TO BE DUPLICATED FIVE TIMES (IN THE CASE OF THE SSDR
;TWENTY-FOUR TIMES.  IT PUTS THE PROPER ADDRESS IN ALL THE REGISTERS THAT
;ARE USED DURING A READ/WRITE TO A BUBBLE.
;
BIOTMP: ;BUBBLE IO TMP LOADER
        PUSH    H
        PUSH    D
        LHLD    BUBNUM
        LXI     D,0
        XCHG
        DAD     D
        DAD     H
        DAD     D
        XCHG
        MVI     A,0C3H  ;JMP OP CODE
        LXI     H,RDBUBD
        DAD     D
        STA     RDDTMP
        SHLD    RDDTMP+1
        LXI     H,WRBUBD
        DAD     D
        STA     WRDTMP
        SHLD    WRDTMP+1
        LXI     H,RDBUBS
        DAD     D
        STA     RDSTMP
        SHLD    RDSTMP+1
        LXI     H,WRBUBC
        DAD     D
        STA     WRCTMP
```

```
        SELD    WRCTMP+1
        POP     D
        POP     E
        RET
;
;
;RDBUBD IS THE ADDRESS SUPPLIED TO RDDTMP WHEN READING THE BUBBLE DATA.
;IT SUPPLIES THE ADDRESS OF THE ACTIVE BUBBLE.
;
RDBUBD:
        IN      80H     ;BUBBLE ONE
        RET
        IN      82H     ;BUBBLE TWO
        RET
        IN      84H     ;BUBBLE THREE
        RET
        IN      86H     ;BUBBLE FOUR
        RET
        IN      88H     ;BUBBLE FIVE
        RET
;
;WRBUBD IS THE ADDRESS SUPPLIED TO WRDTMP DURING A WRITE OPERATION.
;IT SUPPLIES THE ADDRESS OF THE ACTIVE BUBBLE.
;
WRBUBD:
        OUT     80H     ;BUBBLE ONE
        RET
        OUT     82H     ;BUBBLE TWO
        RET
        OUT     84H     ;BUBBLE THREE
        RET
        OUT     86H     ;BUBBLE FOUR
        RET
        OUT     88H     ;BUBBLE FIVE
        RET
;
;RDBUBS SUPPLIES THE ADDRESS TO RDSTMP DURING A STATUS CHECK OF THE
;ACTIVE BUBBLE.
;
RDBUBS:
        IN      81H     ;BUBBLE ONE
        RET
        IN      83H     ;BUBBLE TWO
        RET
        IN      85H     ;BUBBLE THREE
        RET
        IN      87H     ;BUBBLE FOUR
        RET
        IN      89H     ;BUBBLE FIVE
        RET
;
;WRBUBC SUPPLIES THE ADDRESS OF THE ACTIVE BUBBLE TO WRCTMP.  THEN WHEN
;THE COMMAND REGISTERS NEED TO BE WRITTEN TO THE CORRECT BUBBLE IS
;ADDRESSED.
;
```

```
WRBUBC:
        OUT     81H     ;BUBBLE ONE
        RET
        OUT     83H     ;BUBBLE TWO
        RET
        OUT     85H     ;BUBBLE THREE
        RET
        OUT     87H     ;BUBBLE FOUR
        RET
        OUT     89H     ;BUBBLE FIVE
        RET
;
        DS      1
        END
```

LIST OF REFERENCES

1. Get-Away Special (GAS), Small Self-contained Payloads, Goddard Space Flight Center, Special Payloads Division, July 1984.

2. Spiegel, P., "Bubble Memory Applications in Harsh Environments," WESCON/80 Conference Record, v. 17-0/1-2, September 1980.

3. Vail, P.J., "Nuclear Effects on Magnetic Bubble Memories," WESCON/80 Conference Record, v. 17-3/1-4, September 1980.

4. Vail, P.J. and others, "Short Pulse Transient Radiation Effects on a 92 K-bit Magnetic Bubble Memory System," IEEE Transactions on Nuclear Science, v. NS-26, n. 6, pp. 4847-4849, December 1979.

5. Blake, J.B., "Magnetospheric Radiation Environment in a 12-Hour Circular Orbit," AIAA Engineering Notes, v. 18, n. 5, September- October 1981.

6. Fundakowski, Sally W., "Bubbles Bursting with Military Potential," Defense Electronics, July 1984.

7. BPK-5V75A Four-megabit Bubble Memory Prototype Kit User's Manuel, Order Number 2444.001, Intel Corporation.

8. Triebel, W.A. and Chu, A.E., Handbook of Semiconductor and Bubble Memories, Prentice-Hall, Inc., Englewood Cliffs, N.J., 07632, 1982.

9. Memory Components Handbook, Order Number 210830-003, Intel Corporation, 1984.

10. Mokhoff, N., "Magnetic Bubble Memories Making a Comeback," Computer Design, November 1984.

11. Four Megabit Bubble Memory Data Sheets, Order Number 231060-003, Intel Corporation, October 1984.

BIBLIOGRAPHY

Campbell, Bruce A. A Digital Recording System for Space-based Applications Utilizing Four-megabit Magnetic Bubble Memories, M. S. Thesis, Naval Postgraduate School, Monterey, California, June 1985.

Chang, H., Magnetic Bubble Technology: Integrated-Circuit Magnetics for Digital Storage and Processing, IEEE Press, New York, 1975.

Davies, J. E., "Reliability and Environmental Capabilities of the 7110, One-megabit Bubble Memory," WESCON/80 Conference Record, v. 17-4/1-7, September 1980.

Frey, Thomas J., A 32-bit Microprocessor Based Solid State Data Recorder for Space-based Applications, M. S. Thesis, Naval Postgraduate School, Monterey, California, March 1986.

Jordan, Dorsett W., A Matched Filter Algorithm for Acoustic Signal Detection, M. S. Thesis, Naval Postgraduate School, Monterey, California, June 1985.

Wallin, Jay W., Microprocessor Controller with Nonvolatile Memory Implementation, M. S. Thesis, Naval Postgraduate School, Monterey, California, December 1985.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22304-6145	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93943-5002	2
3.	Department Chairman, Code 62 Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, California 93943-5000	2
4.	Commander Naval Space Command Dahlgren, Virginia 22448	2
5.	Professor R. Panholzer, Code 62Pz Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, California 93943-5000	4
6.	Professor M. Cotton, Code 62Cc Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, California 93943-5000	2
7.	Lcdr Sandra L. Coachman, USN 3 Willow Street Newport, Rhode Island 02871	1
8.	Lt Tina-Marie D'Ercole, USN 122 Eaton Avenue Hamilton, Ohio 45013	3
9.	David Rigmaiden 1135 3rd Street Monterey, California 93940	1
10.	Professor S. L. Garrett, Code 61Gx Department of Physics Naval Postgraduate School Monterey, California 93943-5000	1
11.	Professor O. B. Wilson, Jr., Code 61Wl Department of Physics Naval Postgraduate School Monterey, California 93943-5000	1
12.	Space Projects Group, Code 72 Naval Postgraduate School Monterey, California 93943-5000	1
13.	Professor John Simpson Enrico Fermi Institute University of Chicago Chicago, Il 60010	1

14. Captain Roger P. Onorati 1
Commanding Officer
Naval Ocean Research and Development Activity
NSTL, MS 39529

END

DTIC

8-86